Toward a Model-Based Bayesian Theory for Estimating and Recognizing Parameterized 3-D Objects Using Two or More Images Taken from Different Positions

BRUNO CERNUSCHI-FRIAS, MEMBER, IEEE, DAVID B. COOPER, MEMBER, IEEE,
YI-PING HUNG, STUDENT MEMBER, IEEE, AND PETER N. BELHUMEUR, STUDENT MEMBER, IEEE

Abstract—**A new approach is introduced to estimating object surfaces in three-dimensional space from two or more images. A surface of interest here is modeled as a 3-D function known up to the values of a few parameters. Although the approach will work with any parameterization, we model objects as patches of spheres, cylinders, planes, and general quadrics—primitive objects. Primitive surface estimation is treated as the general problem of *maximum likelihood parameter estimation* of the *a priori* unknown primitive surface parameters based on two or more functionally related data sets. In our case, these data sets constitute two or more images taken by cameras at different locations and orientations. A simple geometric explanation is given for the estimation algorithm. Although various techniques can be used to implement this nonlinear estimation, we discuss the use of gradient descent. Experiments are run and discussed. Our approach includes the commonly used stereo approaches as special cases. The Cramer-Rao lower bounds are derived for the achievable error covariance matrices for estimators for the *a priori* unknown parameters. No surface reconstruction can be more accurate than these bounds. The dependence of the bounds on object surface pattern and on the camera and object geometry is shown explicitly. An interesting result arising in this work is that maximum-likelihood estimation of 3-D surfaces also requires maximum likelihood estimation of the pattern on the object surface. Object surface segmentation into primitive object surfaces, and primitive object-type recognition are readily implemented using the probabilistic framework developed in this paper. The attractiveness of our probabilistic formulation is that it now permits a fully Bayesian approach to 3-D surface estimation based on images taken by cameras in two or more positions. For example, recent follow-on papers include estimation of parameterized surfaces based on a large number of images taken by a moving camera [11], [27], estimation of stochastic surfaces based on images taken by cameras in two or more positions [3], and estimation of object surfaces given contour models for the patterns on the surface [28].**

Index Terms—**Bayesian parameter estimation, Cramer-Rao bounds, estimation accuracy, maximum likelihood surface estimation, robot vision, shape from motion, stereo, 3-D shape from multiple images.**

I. Introduction

ESSENTIALLY all 3-D object surface estimation from multiple views to date is based on either active stereo using a laser and one or two cameras for triangulation, or on passive stereo involving matching points in two images and using triangulation, or on optical flow [9], [15], [18], [24], [10]. We suggest a new approach, first presented in [1], [2], in which surfaces of complex objects are approximated by a small number of parameterized 3-D surface patches, and these parameters are estimated from two or more images taken by calibrated cameras from different locations and directions. Note that most manufactured 3-D objects of interest can be well represented by a small number of patches of planes, spheres, cylinders, and cones, and many natural 3-D surfaces are also well represented by these surface patches. (In theory, our approach applies to any *parameterized* surface.) We view the problem within the general context of estimating *a priori* unknown parameters given a data set consisting of two or more functionally related subsets. Estimation accuracy is achieved by processing data in large blocks (large patches of images in our problem). An important contribution of this paper is that we derive the expression for the joint likelihood of two images taken by cameras in different positions. This likelihood is a function of *a priori* unknown parameters, namely, the parameters specifying the 3-D object to be estimated. We then treat 3-D object surface estimation as maximum likelihood estimation, thus using estimators having very desirable properties. However, of greater significance is that being able to express the joint likelihood of data in two or more images as a function of the unknown parameters to be estimated opens the way to treating the estimation of 3-D objects based on these images entirely within a Bayesian framework. This permits the ready solution of many problems. As an example, we derive the Cramer-Rao lower bounds on the error covariance matrix for the estimation of the *a priori* unknown parameters characterizing a primitive 3-D surface patch. No surface reconstruction can be more accurate than these bounds! Other extensions that we have formulated and explored based on Bayesian methods are the maximum *a posteriori* likelihood estimation of surfaces modeled by stochastic processes [3], and the maximum likelihood estimation of surfaces based on a sequence of images taken by a moving camera but using a modest amount of storage and computation [11],

Manuscript received January 16, 1987; revised August 31, 1988. Recommended for acceptance by A. K. Jain. This work was supported in part under NSF Grants ECS-81-19676, INT-86-12525, and IRI-87-15774, under ARO Grant DAA-L03-86-K-0011, the IBM Corporation, the University of Buenos Aires, and Philips Laboratory at Briarcliff, NY.

B. Cernuschi-Frias is with Facultad de Ingenieria, Universidad de Buenos Aires, Buenos Aires, Argentina.

D. B. Cooper, Y.-P. Hung, and P. N. Belhumeur are with the Division of Engineering, Brown University, Providence, RI 02912.

IEEE Log Number 8929335.

[27]. Our approach to the processing of a sequence of images is more general than and has advantages over the use of the Kalman filter. Since we use the joint likelihood for patches of two or more images as a function of the *a priori* unknown 3-D surface parameters to be estimated, other Bayesian problems are easily formulated and implemented. For example, by using sizable patches of images, the form of the dependence of the likelihood function on the parameters to be estimated becomes simple, and minimum probability of error recognizers can be easily implemented for recognizing the shape model for the patch of surface being viewed, i.e., for deciding whether it is a plane, a sphere, a cylinder, etc. (see [12]). Being able to compute the joint likelihood of patches of data from two or more images also permits us to do maximum likelihood segmentation of a complex 3-D surface into primitive surfaces, each specified by a single model. For example, the segmentation can be into smooth surfaces, i.e., those without tangent discontinuities (see [26]). Our approach assumes the use of calibrated cameras. If the camera calibration is unknown, then our approach can still be used but must incorporate the *a priori* unknown camera-model parameter-values as additional parameters to be estimated.

Central to 3-D surface estimation from two (or more) images taken from cameras in different locations and orientations has been the pairing of points from two images that are images of the same point on a 3-D surface. This matching of points in two images is usually done in either of two ways. 1) If the two cameras are physically close and their optical axes are almost parallel, then their images will differ from one another only by translation—one will be a shifted version of the other. Then image 1 can be partitioned into patches, and each patch cross-correlated with image 2 to find its location in image 2. Once this correspondence is known, the location of the surface region in 3-D space seen in the pair of corresponding image patches can be determined by triangulation. Since the surface region seen is usually curved, one would like the patches to be small in order to locate the surface region seen accurately. However, if the images are noisy, large surface patches must be used in order to overcome the effects of the noise, thus introducing some error in determining the correspondences between pairs of points that are in the patch interiors. In addition to this correspondence error, there is always some error in camera calibration. These sources of error can result in appreciable triangulation error because the camera optical axes are almost parallel. 2) An alternative approach that permits a large angle between the camera optical axes to improve triangulation accuracy is to match corresponding small local features in the two images. An example of such a feature is a vertex of a polyhedron or an arc in a boundary. The difficulty here is that a large amount of pattern recognition may be necessary to recognize a pair of corresponding features in the two images.

Among a few interesting new approaches to stereo ranging are the following. Bolles and Baker [5] consider a situation in which a camera moves in a straight line with the direction of the optical axis fixed (in the experiments described), and successive images are taken very close to one another such that a picture differs from its predecessor only by a shift of one or two pixels. Then the images are stacked one in front of the other such that they form a three dimensional array with time being the third axis. The result is that a point on the object surface is now seen as a line or a smooth curve in this three dimensional array, and surface point estimation is accomplished by line or curve estimation in 3-space. Among the potential drawbacks of this approach are that camera motion is restricted, data must be processed at a very high rate, not much information has been presented on the amount of required processing for estimating a suitable number of lines, and no information has been presented on the accuracy of the method. Miller [6] developed an approach where two cameras scan a scene along epipolar lines associated with the same epipolar plane. His problem is to estimate the disparity between the two images, i.e., to match up points on the line in image 2 with points on the line in image 1 that are views of the same surface point. This is accomplished through use of a phase-locked loop. The system has the advantage of potentially working in real time. It has the potential disadvantages that the accuracy may not be very good (no measure is given), occlusion is a problem, initial lock-in may be a problem. Castan and Shen [7] model the distortion from one image to another under the assumption that the surface being viewed is a planar patch. They approximate images locally as second degree polynomials using Taylor series, explore features that are invariant from one image to the other, and use the equations relating the Taylor series expansions at corresponding points in the two images to estimate the planar surface. No mention seems to be made of how they solve the correspondence problem, and other important details are missing, so that it is difficult to appreciate the advantages, disadvantages, and accuracy of the method. Cohen [29] deals with planar surface patches, and uses Markov Random Fields (MRF) to model pattern texture on these surfaces. Planar patch orientation and location is then estimated by comparing the parameters of MRF models fit to pairs of patches one in each image. This elegant work represents a significant, potentially very effective advance in 3-D surface reconstruction, and is in harmony with our concept of the joint estimation of 3-D surface model and surface pattern model in this paper and in [28]. Faugeras, Ayache, and Faverjon [8] develop the idea of estimating points on a 3-D object surface, lines on a surface (and suggest planar surfaces) from a sequence of images. More specifically, they assume that some method has been used to estimate points on a surface based on a pair of images, and assume that the probability distribution for these estimates can be determined. They then assume that a sequence of such estimates and associated distribution are known for a sequence of images. Their contribution, then, is to use the extended Kalman filter for combining this sequence of estimates to obtain improved estimates of the surface points. They derive the equations for estimating points and lines, and suggest that it can be extended to planes. Among the errors they take into account, are those in camera calibration. Their concept is important, though they do not tackle here the problem of initially optimally estimating the surface points or lines from a pair of images. Ohta and Kanade [4] extend an approach of Baker but use intervals between edges instead of edges themselves, and apply the dynamic programming technique performing both the intra- and the inter-scanline search simultaneously for matching points in the two images. Waxman and Wohn [10] take a different approach, focusing on the use of optical flow to estimate the normals of 3-D surfaces by solving some local equations. To estimate position, they suggest using binocular optical flow. Being able to extract information by analytically solving equations is attractive. On the other hand, 3-D surface estimation based on optical flow is less accurate than the use of stereo because of the small baseline. Last, we mention a recent paper by Eastman and Waxman [25]. They use two cameras having parallel optical axes and locally model the disparity function as a quadric. Then, upon matching corresponding contours in the two images, they can solve for the *a priori* unknown parameters in the quadratic disparity function, and from this can obtain an estimate of a local quadric approximation to the 3-D surface depth function. This has some similarity to our estimation algorithm in Section II (presented earlier [2]). There are significant differences. Their approach has the desirable feature that once contour matching has been accomplished, estimation of the disparity function locally is computationally simple. Some positive features of our approach are the following. We estimate sizable 3-D surfaces directly, and these surfaces can be arbitrary parameterized or stochastic surfaces. Camera relative positions can be completely arbitrary. The computational complexity of matching up pairs of contours—one contour in each image—is avoided; instead we search in the surface parameter space (which could also be costly, but can be done with simple parallel processing). Last, since our formulation is a Bayesian one we can make optimal use of probabilistic prior information concerning the surfaces to be estimated. Our paper is an expansion of one where our 3-D surface estimation algorithm was first proposed [1], [2]. The preceding papers have been directed at estimating points, curves, planes, cylinders, spheres, or general parameterized surfaces in 3-D space. In general for making inferences about 3-D surfaces irrespective of the type of sensing used, the idea of Besl and Jain [19] and Bolle and Sabbah [20] for the representation of surface patches by their Gaussian and mean curvatures appears to be a very useful generalization.

Sections II-A–II-F introduce our 3-D surface estimation algorithm, show examples of its use with planar, spherical, and cylindrical surfaces, and show that the estimation is maximum likelihood estimation. The shape of the likelihood function, as the parameters to be estimated are varied, is explored. The algorithm in Section II-D, based on the use of a sequence of images, is a computationally simple way around the problem of having to minimize a multimodal performance functional. Experiments are shown with both partially artificially generated data, and with real data taken by a moving camera. Section II-F deals with a fundamental complication arising from the quantization of images into pixels. To this point, the focus has been on the imaging geometry, an algorithm for 3-D surface estimation, and experiments.

Sections III and III-A are devoted to the derivation of the Cramer-Rao Lower Bounds on the error covariances for the *a priori* unknown parameters to be estimated for the object models used. This involves working through some algebra, but is important because in 25 or more years of published literature on computer vision, there are almost no upper or lower bounds on achievable estimation accuracy tied to the raw image data. The only published results we are aware of are [21] and [22], [23]. The present problem is sufficiently general that derivation of the C.R. Bounds here provides insight into how they may be applied elsewhere. For those not interested in the derivation of the bounds, the final expressions are given by Eqs. (37) and (41), and pertinent discussion is given in Section III-A. Section III-B and III-C provide a physical interpretation of the Bound, showing the explicit dependence on camera geometry, object surface pattern, and object surface parameterization. And Section III-D contains an example of a numerical computation of the Bound.

A. Notation and Description of Camera Motion

Let P be a point in 3-D space and $\mathbf{r}'^T = (x', y', z')$ be its representation in the fixed orthogonal world reference frame.[1] Since we assume that objects do not move, this reference frame is fixed with respect to the objects viewed by the camera, and we will call it the object reference frame (ORF). Let $\mathbf{r}^T = (x,y,z)$ be the representation, of the point P, in the orthogonal reference frame attached to the camera. This reference frame, the camera reference frame (CRF), is such that: 1) The camera optical axis is parallel to the z axis, and it looks at the negative z axis. 2) The x and y axes are parallel to the sides of the image. 3) The origin of the camera reference frame coincides with the center of the image plane. The image is corrected so that the view is not inverted top to bottom and left to right, i.e., a central projection is used.

Let B denote the 3×3 orthogonal rotation matrix that specifies the three unit coordinate vectors for the CRF in terms of the three unit coordinate vectors for the ORF. Let $\mathbf{r}_c'$ specify the origin of the CRF in the ORF. Then

$$\mathbf{r} = B^T(\mathbf{r}' - \mathbf{r}_c'), \quad \text{and} \quad \mathbf{r}' = B\mathbf{r} + \mathbf{r}_c'. \qquad (1)$$

Note that the rotation matrix B and the translation vector $\mathbf{r}_c'$ are assumed to be known.

[1] A symbol in boldface is a column vector, a superscript capital T attached to a vector denotes vector transpose.

B. Images of an Object Surface Point in Two Image Frames

Fig. 1 illustrates the *orthographic* imaging model [15] used, i.e., all rays from points on the object surface to the image plane are roughly parallel. This model, an approximation to the pinhole model (perspective projection), is explained in Appendix A. It is applicable when the object diameter is small compared with object to camera distance. With slight modification, all of our algorithms can be run using the pinhole model, and the experiment using real images taken by a moving camera, shown at the end of Section II-D, used the pinhole model in the surface estimation. Let P denote a point on a parameterized 3-D surface of interest. Point P on the object surface is seen as points having coordinates $\boldsymbol{s}$ and $\boldsymbol{u}$ in images 1 and 2, respectively. We assume a *Lambertian reflectance model*. Then the images of point P at $\boldsymbol{s}$ and $\boldsymbol{u}$ will have the same intensity. The techniques proposed will not apply without modification to specular reflectors because the location of points on the object surface at which specular reflection occurs depends on the camera location. Since most surfaces of interest are largely Lambertian, the assumption is a useful one. Hence,

$$I_2(\boldsymbol{u}) = I_1(\boldsymbol{s}) \tag{2}$$

where $I_1(\boldsymbol{u})$ and $I_2(\boldsymbol{s})$ are the picture functions (image intensity functions) in Frames 1 and 2, respectively. For those cases where the Lambertian assumption does not apply, a possible modified approach is to use an edge map. Here, pixels are given values of 128 or 0 depending on whether they are detected as being edge points or nonedge points, respectively. These maps are then smoothed to obtain more continuous arrays, and these are used as though they are regular picture functions in our estimation algorithms. The usefulness of the edge map is that it is a representation of rapid changes in the object surface patterns, and largely unaffected by the presence of some specular component in the object surface. See [24] for other approaches to partially specular surfaces.

Our approach is applicable to any parameterized surface. However, in this paper we illustrate the approach through the use of planar, cylindrical, spherical, and general quadric primitive surfaces. The equation of the general quadric surface is given by (3). The other three primitive quadrics are obtained by imposing suitable constraints among the coefficients in (3).

$$a_{11}x^2 + 2a_{12}xy + 2a_{13}xz + a_{22}y^2 + 2a_{23}yz + a_{33}z^2 + 2a_{14}x + 2a_{24}y + 2a_{34}z + a_{44} = 0. \tag{3}$$

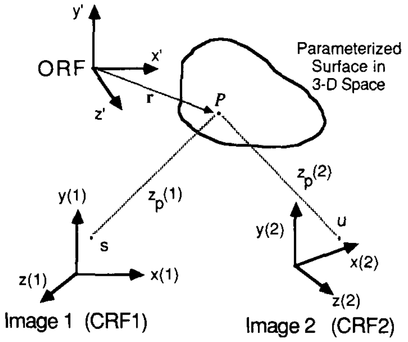


Fig. 1. Images of an object surface point in two image frames.

These surfaces are described in the ORF which we take to be CRF1, i.e., the first camera reference frame, and are uniquely determined by specifying the values of a parameter vector $\boldsymbol{a}$. Denote $\boldsymbol{a}^T = (a_{11}, a_{12}, \cdots, a_{44})$. For a general surface, $\boldsymbol{a}$ will have K components. Thus, the image at point $\boldsymbol{s}^T = (x(1), y(1))$ in the first image frame is the image of a point having coordinates $x(1)$, $y(1)$, $z(1)$ on the object surface where $z(1)$ is the solution of (3) when $x(1)$ and $y(1)$ are specified. We see that the picture function at point $\boldsymbol{s}$ in CRF1 is the picture function at point $\boldsymbol{u}$ in CRF2 where

$$(\boldsymbol{u}^T, z(2))^T = B^T(\boldsymbol{r}(1) - \boldsymbol{r}_c(2)) \tag{4}$$

and $\boldsymbol{u}^T = (x(2), y(2))$. Parameters specifying this transformation are three rotation angles specifying the rotation matrix B, and three translation components specifying the location $\boldsymbol{r}_c(2)$ of the origin of CRF2 in CRF1. Denote this six component parameter vector by $\boldsymbol{b}$. Denote the functional relationship between $\boldsymbol{s}$ and $\boldsymbol{u}$ by (5).

$$\boldsymbol{u} = \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a})). \tag{5}$$

Note $\boldsymbol{u}$ depends explicitly on $\boldsymbol{s}$ and $z(1)$, and $z(1)$ is determined by both $\boldsymbol{s}$ and $\boldsymbol{a}$ as shown in (3).

C. On the Form of $\boldsymbol{u} = \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a}))$

Assume we have a rigid surface in 3-D space. The equation of this surface with respect to the object reference frame is

$$g(\boldsymbol{r}(1)) = 0 \tag{6}$$

and for CRF2, upon using (4),

$$g(B(2)\,\boldsymbol{r}(2) + \boldsymbol{r}_c(2)) = 0. \tag{7}$$

Denote

$$C = B^T(2) \tag{8}$$

and

$$\boldsymbol{d} = -B^T(2)\,\boldsymbol{r}_c(2). \tag{9}$$

Then

$$\boldsymbol{r}(2) = C\boldsymbol{r}(1) + \boldsymbol{d} \tag{10}$$

We partition the C matrix as:

$$C = \begin{bmatrix} C_{11} & \boldsymbol{c}_{12} \\ \boldsymbol{c}_{21} & c_{22} \end{bmatrix} \tag{11}$$

where C_{11} is 2×2, $\boldsymbol{c}_{12}$ and $\boldsymbol{c}_{21}^T$ are 2×1 and c_{22} is a number. Partition $\boldsymbol{d}$ as $\boldsymbol{d}^T = (\boldsymbol{e}^T d_3)$ where $\boldsymbol{e}$ is 2×1 and d_3 is a number.

Then from the preceding

$$u(2) = C_{11}u(1) + c_{12}z(1) + e. \qquad (12)$$

Equation (12) is the parametrized equation of a line. That is, a point with coordinates $u^T(1) = (x(1), y(1))$ in the first frame is the image of some point in the surface. Equation (12) gives the locus of possible locations (parametrized by the unknown $z(1)$) that the point may have in frame 2. This is the epipolar line in image plane 2 associated with point $u(1)$ in image plane 1.

II. Estimation of Surface Parameters a Using Two Images

If b is known and a is a_T, the true a, then

$$I_1(s) = I_2\big(h(s, b, z(s, a_T))\big) \qquad (13)$$

for each s. Choose a square $M \times M$ pixel window in CRF1. Denote this pixel set by D. Consider the error measure

$$e_D(a) = M^{-2} \sum_{s \in D} \big[I_1(s) - I_2\big(h(s, b, z(s, a))\big)\big]^2. \qquad (14)$$

Then $e_D(a)$ is a minimum at $a = a_T$. Our problem is to estimate a_T by minimizing (14) with respect to a. An interpretation of (13) is that the image $I_2(u)$ can be transformed into the image $I_1(s)$ by a varying scale change that locally consists of a rotation, a nonisotropic stretching, and a translation.

The minimization technique we have used is gradient descent. The gradient of (14) is

$$\frac{\partial e_D}{\partial a} = -2M^{-2} \sum_{s \in D} \big[I_1(s) - I_2(u)\big] \frac{\partial I_2(u)}{\partial a} \qquad (15)$$

where from (12)

$$u = h\big(s, b, z(s, a)\big) = C_{11}s + c_{12}z(s, a) + e. \qquad (16)$$

Equation (15) is an explicit function of s, b, z, and is implicitly dependent on a because of the determination of z by s and a. Use of the chain rule gives[2]

$$\frac{\partial I_2(u)}{\partial a} = \frac{\partial I_2(u)}{\partial u} \frac{\partial h\big(s, b, z(s, a)\big)}{\partial a}$$

$$= \frac{\partial I_2(u)}{\partial u} \frac{\partial h\big(s, b, z(s, a)\big)}{\partial z} \frac{\partial z(s, a)}{\partial a} \qquad (17)$$

with $\partial h\ (s, b, z(s, a))/\partial z = c_{12}$. In general, it may be inconvenient to express z as an explicit function of a. In such cases to proceed further denote the left side of (6) by $g(x, y, z, a)$. Then (6), the equation of the object surface, is $g(x, y, z, a) = 0$. Hence, we can write

$$\frac{\partial z(s, a)}{\partial a} = -\frac{\partial g(x, y, z, a)}{\partial a} \Big/ \frac{\partial g(x, y, z, a)}{\partial z}. \qquad (18)$$

Putting (16)–(18) together results in

$$\frac{\partial e_D(a)}{\partial a} = 2M^{-2} \sum_{s \in D} \big[I_1(s) - I_2(u)\big] \left[\frac{\partial I_2(u)}{\partial u}\right] \frac{\partial h(s, b, z)}{\partial z}$$

$$\cdot \left[\frac{\partial g(x, y, z, a)}{\partial a} \Big/ \frac{\partial g(x, y, z, a)}{\partial z}\right]. \qquad (19)$$

Hence, computation of (19) involves processing the data to obtain $I_1(s)$, $I_2(u)$, and $\partial I_2(u)/\partial u$, and computing c_{12} and the last two partial derivatives from the known camera motion and knowledge of $g(x, y, z, a)$.

A steepest descent algorithm for minimizing (14) is

$$a_{n+1} = a_n - \frac{\partial e_D(a_n)}{\partial a} \Delta_n. \qquad (20)$$

We use a Δ_n that depends on $e_D(a_n)$ and $\partial e_D(a_n)/\partial a$ and has magnitude that goes to 0 as n goes to infinity.

A. Algorithm Operation-Interpretation, and Experiments with a Sphere

To illustrate the approach, consider a spherical surface described by the equation

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2. \qquad (21)$$

For this surface, z can be solved for explicitly, via

$$z = z_0 \pm \big(R^2 - (x - x_0)^2 - (y - y_0)^2\big)^{1/2}. \qquad (22)$$

The positive square root is used since the outside surface of the sphere is seen by the camera looking in the negative z direction. Hence,

$$\left(\frac{\partial z(s, a)}{\partial a}\right) = \left(\frac{\partial z}{\partial x_0}, \frac{\partial z}{\partial y_0}, \frac{\partial z}{\partial z_0}, \frac{\partial z}{\partial R}\right)$$

$$= \big((x - x_0)/(z - z_0), (y - y_0)/(z - z_0),$$

$$1, R/(z - z_0)\big) \qquad (23)$$

and $z - z_0 = (R^2 - (x - x_0)^2 - (y - y_0)^2)^{1/2}$. The vector $\partial z/\partial a$ can be computed directly from this.

The analogous equations for planes, cylinders and general quadrics are presented in Appendixes C, D, and E, respectively.

Fig. 2 is useful for illustrating, in two dimensions, the operation of our algorithm for estimating a_T. Spheres in 3-D are shown as circles. Consider the processing of the image patch between points s' and s'' in Frame 1. This patch is the image of the patch between points p' and p'' on the true sphere labeled a_T. The same patch on the sphere surface gives rise to the image patch between points u' and u'' in Frame 2. Now suppose the system's estimation of a_T is $\tilde{a}$. The associated sphere is shown. The performance functional for the estimate of a is given by (14) and is computed as follows. The system thinks that the locations on the sphere surface that give rise to the images at points s' and s'' in Frame 1 are the intersections of the dashed lines, from s' and s'', with the sphere labeled $\tilde{a}$. These sphere surface points would be seen as the images at point $\tilde{u}'$ and $\tilde{u}''$ in Frame 2. Hence, the system takes the image patch between points $\tilde{u}'$ and $\tilde{u}''$ in Frame

[2] The notation used here is that $\partial I_2(u)/\partial a$ is a K component row vector, and $\partial h(s, b, z(s, a))/\partial a$ is a $2 \times K$ matrix.

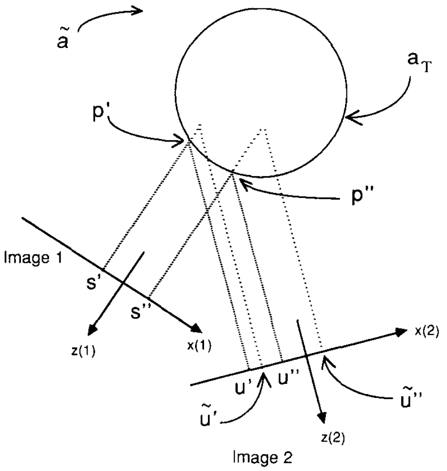


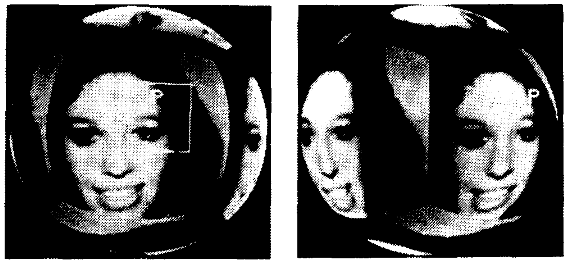
Fig. 2. Two-dimensional illustration of the geometry for the surface estimation algorithm.

2 and assumes that the image at each point $\boldsymbol{u}$ in this interval is the same image as the image at a point $\boldsymbol{s}$ in the interval between s' and s'' in Frame 1. The points $\boldsymbol{u}$ and $\boldsymbol{s}$ are related geometrically as in the figure, or algebraically by (4). Performance functional (14) requires computing the error $I_1(\boldsymbol{s}) - I_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a})))$. Of course, in the absence of image noise, (14) is zero only when $\boldsymbol{a} = \boldsymbol{a}_T$.

We make the following interesting observations. From the geometry of image formation in Fig. 2, the varying scale change that maps the image patch over interval $[s', s'']$ in Frame 1 into the image patch over interval $[\boldsymbol{u}', \boldsymbol{u}'']$ is seen. Note that both a scale change and a translation are involved in this 2-D illustration.

If the incorrect $\boldsymbol{a}$ is used in computing the performance functional (14), the patch of image used in Frame 2 is that over the interval $[\tilde{\boldsymbol{u}}', \tilde{\boldsymbol{u}}'']$. Note that this interval is both a shift and a varying scaling of the interval $[\boldsymbol{u}', \boldsymbol{u}'']$. If instead of a sphere, we were dealing with a planar surface, the scale change would be constant throughout the image.

The interpretation is further illustrated by the following 3-D computer simulations. Fig. 3(a) and (b) are two frames illustrating images of the same sphere but taken at different locations and orientations. The angle between the optical axes of the two cameras is 45°. The data was generated by taking a real image with a T.V. camera and projecting the image pattern onto a sphere, from a few different directions. The pattern projected onto the sphere in this way is the pattern that is then viewed by the two cameras to create Frames 1 and 2. Note that all these projections are done by computer simulation. The image patch used in Frame 1 is the interior of the square shown there. Point P is its center point. Points P in Frames 1 and 2 are locations of images of the same point on the 3-D sphere surface. Note how the image in the vicinity of Point P in Frame 2 is a varying scaled transformation of the image in the vicinity of Point P in Frame 1. In Fig. 4, we refer to the six image boxes as 1 through 6 starting with the upper left and moving left to right and top to bottom. Box 1 is the image shown in the square window in Fig. 3(a). The system begins with a guess as to $\boldsymbol{a}$. This initial guess for $\boldsymbol{a}$ is shown as that associated with Box 2 in Table I. Using this $\boldsymbol{a}$, for each point $\boldsymbol{s}$ in the window in Fig. 3(a) the system takes the image at point $\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a}))$ in Fig. 3(b) and puts this picture function value at location $\boldsymbol{s}$ in an array. The image so formed is that in Box 2. It is the difference of the picture functions of these two images that enters as $I_1(\boldsymbol{s}) - I_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a})))$ in (14). (If this $\boldsymbol{a}$ were equal to $\boldsymbol{a}_T$, then in the absence of image noise, the image in Box 2 would be identical to that in Box 1.) Box 3 is the image formed in this way using $\boldsymbol{a}_1$, the parameter vector following the first iteration of our param-



(a) (b)

Fig. 3. Two computer generated images of a sphere taken at different locations and orientations. (Section II-A.)

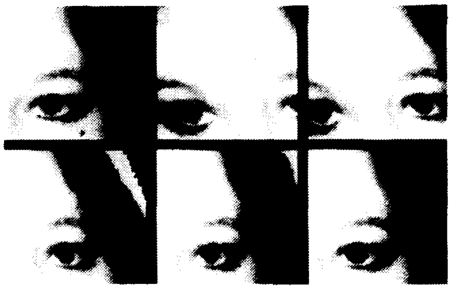


Fig. 4. Image boxes are numbered 1 through 6, running left to right, top to bottom. Box 1 is taken by camera 1. Box k (k = 2, 3, $\cdots$, 6) is associated with the estimate of $\boldsymbol{a}$ in line k of Table I.

TABLE I

Image Box	a			
	x_0	z_0	z_0	R
2	40.00	40.00	-2040.00	128
3	20.83	40.37	-2016.74	128
4	20.79	40.40	-2004.72	128
5	10.52	30.30	-2009.08	128
6	0.41	1.13	-1999.89	128

$\mathbf{a}_T$	0.00	0.00	-2000.00	128
$\hat{\mathbf{a}}$	0.41	1.13	-1999.89	128

eter estimation algorithm. In Boxes 4 and 5 are the transformed images associated with $\boldsymbol{a}$ values at intermediate stages of the estimation process. Finally, Box 6 is the image associated with the $\boldsymbol{a}$ value found at the last estimation stage. These $\boldsymbol{a}$ values for the stages associated with the six boxes are shown in Table I. If the final estimate $\hat{\boldsymbol{a}}$ is equal to $\boldsymbol{a}_T$, then in the absence of image noise, the images in boxes 1 and 6 would be identical.

Figs. 5 and 6 show the shape of the error function $e_D(\boldsymbol{a})$ for the experiment described. Fig. 5 shows the graph of the error function produced by holding z_0 and R fixed at the true values and varying x_0 and y_0 over the ranges $-150 < x_0 < 200$ and $-170 < y_0 < 170$ where $\boldsymbol{a}_T$ is given in Table I. Fig. 6 shows the same error function with x_0 and R held fixed and y_0 and z_0 varied over the ranges $-90 < y_0 < 120$ and $-2120 < z_0 < -1880$. Note that the error function changes much more rapidly with change in z_0 than with change in x_0 or y_0 where z_0 is the distance from the sphere center to the camera.

B. Experiments with a Cylinder when Using Two Images

There are a number of possible parameterization for the cylinder. The following one has been found to be desirable for use in the algorithm given in (20). Consider Fig. 7. Then the cylinder axis location and orientation are specified by the parameter vector (x_0, z_0, θ, ϕ). We arbitrarily choose y_0 to be any point in the vicinity of the image patch being processed in CRF1. Then x_0 and z_0 are the remaining coordinates to be determined for specifying a point on the cylinder axis. As can be seen in Fig. 7, θ and ϕ are the angles specifying the cylinder axis orientation, with θ being the angle between the z axis and the projection of the cylinder axis into the xz plane, and ϕ being the angle between the y axis and the cylinder axis.

Fig. 8(a) and (b) are two frames illustrating images of the same cylinder but taken at different camera locations and orientations. The angle between the optical axes of the two cameras is 45°. The data were generated and the images were formed in the same way as described in Section II-A for the sphere. The image patch used in image 1 is the interior of the square shown there in Fig. 8(a). The portion of the cylinder surface seen in this square patch, is seen as the patch in the four sided polygon in image 2 shown in Fig. 8(b). (Observe that there is a dashed line along the left border of the cylinder image in Fig. 8(b). This is due to spatial quantization and the fact that the border occurs at the image of a portion of the surface pattern where there is a discontinuity from a white region to a dark region in the pattern intensity.) In Fig. 9, we refer to the six image boxes as 1–6 starting with the upper left and moving left to right and top to bottom. The pictures shown are similar to those in Section II-A for a sphere. The system begins with a guess for $\boldsymbol{a}$. This initial guess is given in line 1 in Table II. As in Section II-A, using this $\boldsymbol{a}$, for each $s \in D$ the system takes $I_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}))$ and puts it at location s in an array, thus generating the image shown in Box 1. Boxes 2–5 are images formed in this way for a subset of the sequence of values for $\boldsymbol{a}$ occurring in the iterative estimation of $\boldsymbol{a}_T$. The image constructed in this way for the final estimate of $\boldsymbol{a}_T$ is shown in Box 5. Box 6 is the image constructed given the true

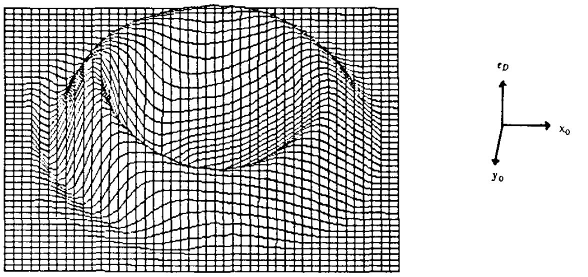


Fig. 5. Error function produced by holding z_0 and R fixed at the true values and varying x_0 and y_0 over the ranges $-150 < x_0 < 200$ and $-170 < y_0 < 170$.

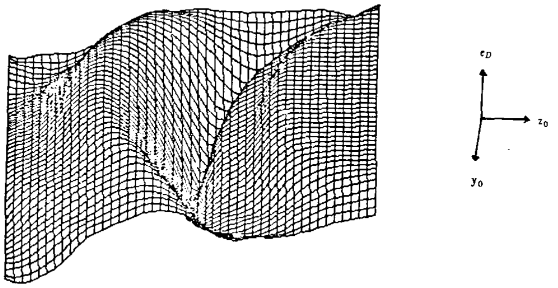


Fig. 6. Error function with x_0 and R held fixed and y_0 and z_0 varied over the ranges $-90 < y_0 < 120$ and $-2120 < z_0 < 1880$.

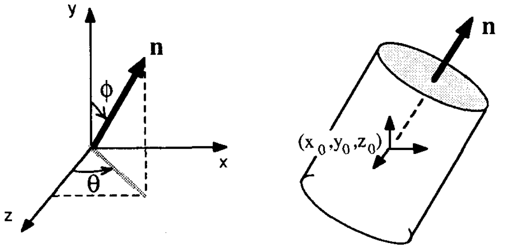
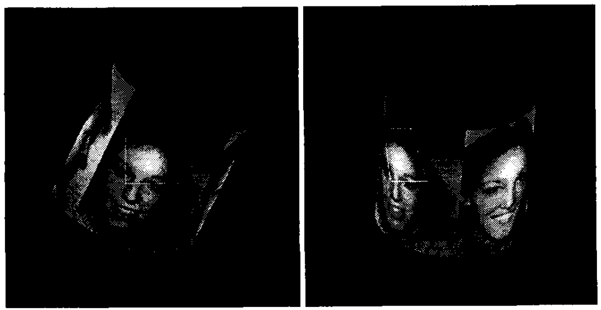


Fig. 7. Parameterization of cylindrical surfaces used with the experiments in Section II-B.



(a) (b)

Fig. 8. Two computer generated images of a cylinder taken at different locations and orientations. (Section II-B.)

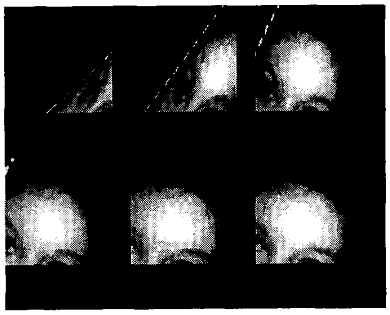


Fig. 9. Image boxes are numbered 1 through 6, running left to right, top to bottom. Box k ($k = 1, 2, \cdots, 6$) is associated with the estimate of $\boldsymbol{a}$ in line k of Table I.

TABLE II

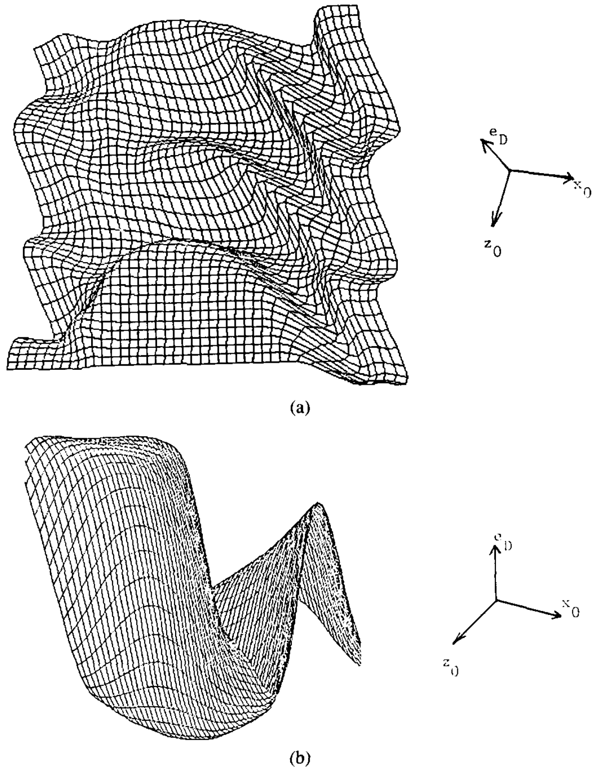
Image Box	a			
	x_0	z_0	θ	ϕ
1	-40.0	-1960.0	30.0	50.0
2	-38.9	-1968.7	36.0	38.2
3	-24.8	-1994.5	43.2	23.5
4	-11.6	-2000.2	47.0	27.1
5	-0.7	-2000.2	50.9	31.7
6	0.0	-2000.0	53.7	31.6

surface parameters. This is the same patch as marked in Fig. 8(a). Note that the images in Boxes 5 and 6 are almost identical, as they should be. The parameter estimates are shown in Table II. The final estimates are close to the true values.

The error as a function of x_0, z_0 in the vicinity of $\boldsymbol{a}_T$ is shown in Fig. 10. The global minimum is in the first valley in the figure. The somewhat multimodal error function plotted is 400 by 400 units, for a cylinder radius of 128 units. An enlargement of this function in the vicinity of $\boldsymbol{a}_T$ is shown in Fig. 10(b). The plot shown is 160 by 160 units. Notice that the function changes rapidly in the z_0 direction, but much more slowly in the x_0 direction. Fig. 11(a) is a plot of the error as a function of θ and ϕ. The extent of the plot is 400 degrees in each direction. Fig. 11(b) is a 40 by 40 degree plot of a portion about $\boldsymbol{a}_T$. Note that there is slow variation as a function of ϕ. The reason is that for the geometry of this example, the object surface patch involved is *roughly* parallel to the plane that the cylinder axis moves in when ϕ is varied but θ is held fixed. Hence, the geometry is much like that of a planar surface being moved in a plane parallel to the surface. The error function that we are using is insensitive to such motion, i.e., to such parameter variation. The error function does change rapidly for parameter variation in other directions, e.g., the direction of θ for the specific $\boldsymbol{a}_T$ involved here.

C. Experiments with a Cylinder when Using Two Edge Maps

The preceding theory is predicted on the assumption of a Lambertian surface, i.e., that a point on the object surface appear with the same image intensity no matter what the location and orientation of the camera be. Now most surfaces have some specular (mirror) component. If the Lambertian assumption does not apply, then an edge detector can be run over the image, and pixels located at large discontinuities in the picture function are given a fixed large value. All other pixels are given value 0. Then this resulting array is low-pass filtered to produce what we are calling the edge map. This edge map should be a function only of the pattern on the object surface, and not of the illumination nor of the reflection properties of the surface, except for the rare situation of a very highly specular surface reflecting an illumination intensity pattern having sharp large spatial discontinuities. We treat this edge map as a regular image and use it in our algorithm. Fig. 12(a) and (b) are obtained by edge detection of Fig. 8(a) and (b), respectively. They are then low pass filtered to obtain the edge maps used in the surface reconstruction algorithm. Table III lists the estimates of $\boldsymbol{a}_T$ at a number of iterations in the algorithm, and Fig. 13 is a sequence of reconstructions analogous to those in Fig. 9. The minimum of (14) as a function of $\boldsymbol{a}$ is larger when using the edge maps rather than the images in Fig. 8(a) and (b). The reason is that the edges associated with a point on the object surface have different widths in Fig. 8(a) and (b). Hence, even in the absence of sensor noise, this performance functional will not be zero when $\boldsymbol{a} = \boldsymbol{a}_T$. Nevertheless, the estimation of $\boldsymbol{a}_T$ based on the edge map data does work reasonably well.



(a)

(b)

Fig. 10. (a) Plot of the error function as a function of x_0 and z_0 centered at $\boldsymbol{a}_T$. (b) An enlargement of the above error function in the vicinity of $\boldsymbol{a}_T$.

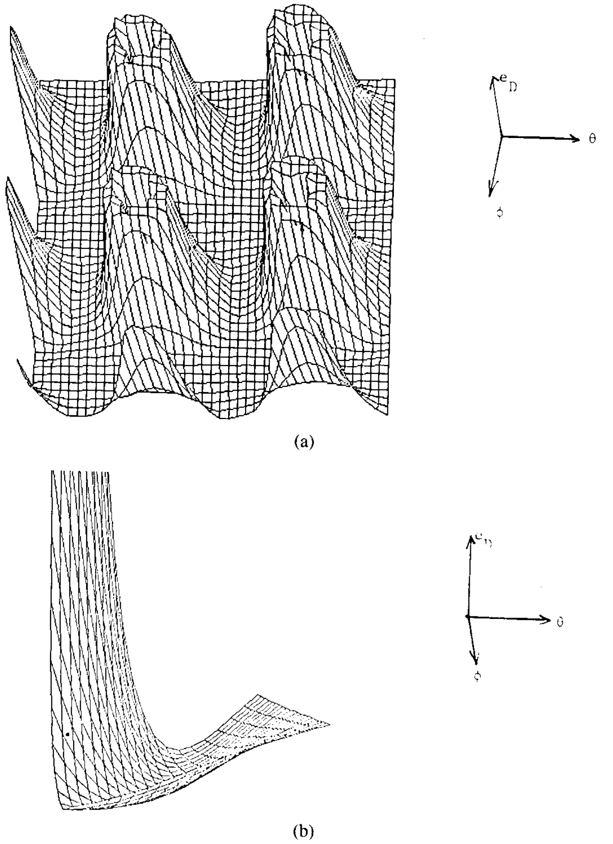


Fig. 11. (a) Plot of the error function as a function of θ and ϕ centered at a_T. (b) An enlargement of the above error function in the vicinity of a_T.

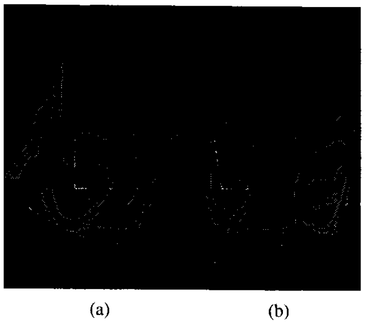


Fig. 12. Edge maps obtained by simple edge detection of Fig. 8(a) and (b). (Section II-C.)

TABLE III

Image Box	a			
	x_0	z_0	θ	φ
1	40.0	1980.0	70.0	15.0
2	35.7	-1993.9	64.5	20.8
3	27.9	-1997.3	59.2	26.4
4	20.6	-1997.4	55.8	28.3
5	13.7	-1999.1	51.7	28.4
6	0.6	-2000.1	48.3	28.1
7	0.0	-2000.0	53.7	31.6

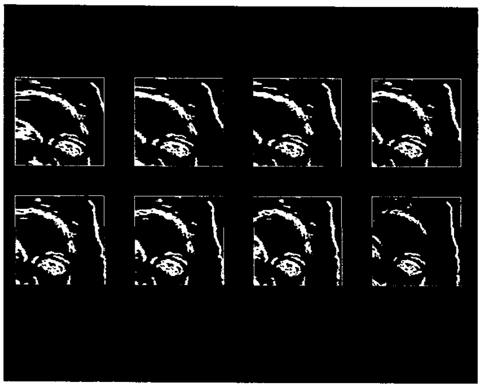


Fig. 13. Image boxes are numbered 1 through 8, running left to right, top to bottom. Box k ($k = 1, 2, \cdots, 7$) is associated with the estimate of $\boldsymbol{a}$ in line k of Table III. Box 8 is the marked window shown in Fig. 12(a).

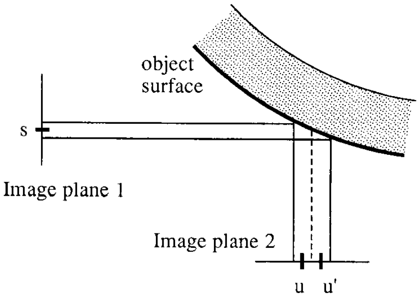


Fig. 14. Two pixels in image plane 2 may correspond to one pixel in image plane 1, as shown here.

D. Experiments with a Cube Having Planar Surfaces

Two things of interest are illustrated here. First, image 1, the image of a cube, is partitioned into small windows, and the behavior of the estimates of the surface patches seen within these windows is studied. Some windows see portions of only one planar surface, and some see portions of two, or even three, planar surfaces. Second, the error function (14) is unimodal but shallow over a large region in the 3-D parameter space when the angle between the camera optical axes is small, e.g., 1°. And the function is multimodal over this region but with a narrow valley about the true parameter value when the angle between the camera optical axes is large. This behavior is exploited to arrive at a computationally simple search algorithm for 3-D surface parameter estimation by using a sequence of images (four in this example), where the increasing angles between the optical axis of the first camera and those of successive ones take the values 1°, 3°, 10°.

Fig. 15(a) shows image 1, a computer generated image of a corner of a cube, partitioned into many small windows. A plane is specified by the normal vector $\boldsymbol{n}$, and the distance d in the direction of the z axis, from the center of the image window to the planar patch. We specify the normal vector by the two angles, ψ and ρ, as shown in Fig. 16. Then, the equation for the plane with respect

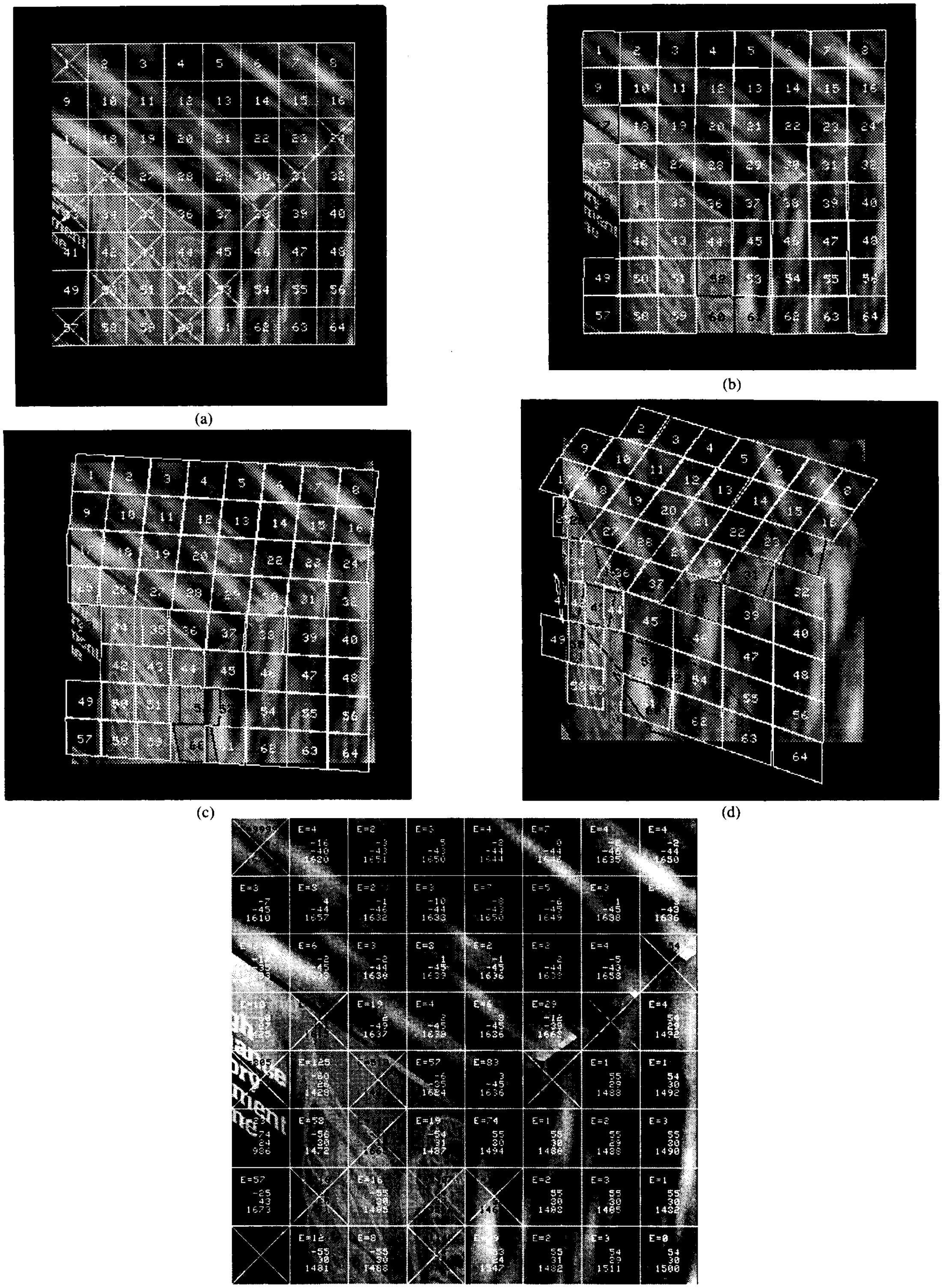


Fig. 15. (a) shows image 1, a computer generated image of a corner of a cube, partitioned into many small windows. (b) shows image 2 marked with regions corresponding to the windows in image 1, using the estimates based on image 1 and image 2. (c) shows image 3 marked with regions corresponding to the windows in image 1, using the estimates based on image 1 and image 3. (d) shows image 4 marked with region corresponding to the windows in image 1, using the estimates based on image 1 and image 4. (e) shows the values of the estimated parameters based on image 1 and image 4.

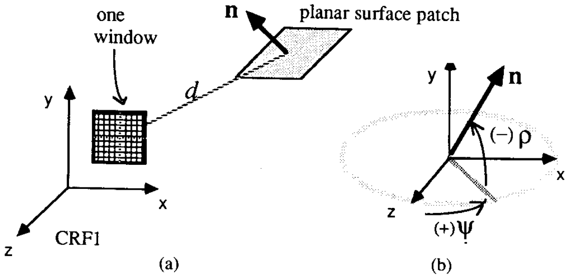


Fig. 16. Parameterization of planar surface used in Section II-E.

to CRF1, which is the ORF, is

$$z = (-\tan \psi)(x - x_0) + \left(\frac{\tan \rho}{\cos \psi}\right)(y - y_0) + (-d)$$

or

$$z = (-\tan \psi)x + \left(\frac{\tan \rho}{\cos \psi}\right)y + (-d')$$

where x_0 and y_0 are the 2-D coordinates of the center of the chosen window and d is a function of x_0 and y_0. The distance d' here is the distance along the z axis from the origin to the plane. In order to compare the parameter values of different surface patches, let the vector $\boldsymbol{a}$ denote the three parameters, ψ, ρ, and d'. Since the image of the cube is generated by the computer, we know, for example, that the true parameter $\boldsymbol{a}_T$ for the top plane of the cube is ($\psi = 0°$, $\rho = -45°$, $d' = 1638$).

As mentioned, the error function to be minimized, (14), is smooth and unimodal over a large region in object parameter space when the angle between opitcal axes and the difference in locations of the two cameras are small. This occurs because for a small baseline, the stereo disparity changes slowly with change in 3-D surface parameters. However, when the baseline is large, (14) becomes multimodal with a narrow valley about $\boldsymbol{a}_T$. Since we have a sequence of images available, we can begin with a small baseline image pair. We can start the gradient descent virtuely anywhere in the parameter space. Because of the unimodality of (14), the algorithm quickly converges to the minimum of the valley. The estimate of $\boldsymbol{a}_T$ found by the algorithm will not be accurate, because the valley is broad. But this estimate will lie within the valley containing $\boldsymbol{a}_T$ of (14) for a longer baseline geometry, and can therefore be used as the starting value for a new estimation of $\boldsymbol{a}_T$ based on image 1 and image 3. The process is repeated again, etc. Four images, Fig. 15(a)–(d), with the second, third, and fourth optical axes making angles 1°, 3°, 10° with the first optical axis, are used in the experiment. The initial parameter value used in the estimation algorithm is ($\psi = 0.0$, $\rho = 0.0$, $d = 2000$) for all the windows in the image containing the three sides of the cube. That is, we start with an initially guessed plane that is perpendicular to the optical axis of the first camera position, and allow it to tilt, rotate, and shift to the true position by searching for the best match in the parameter space. As mentioned before, d is the distance from the center of the image window to the planar patch in the direction of the z axis. The first ımage is partitioned into 64 square windows, each with 64 × 64 pixels, shown in Fig. 15(a). The algorithm is run on the first two images, independently for each window in Fig. 15(a). Based on the convergent surface parameters, the regions in image 2 corresponding to the windows in image 1 are drawn in Fig. 15(b) for illustration. Windows numbered in white are those for which the estimated plane is consistent with the two images, i.e., (14) is small, whereas windows numbered in black are those for which such consistency is absent, i.e., (14) is large. A threshold was *arbitrarily* set at $E = 60$ for this determination. Here, most of the convergent parameter estimates are still quite far from the true surface parameter values, but are much closer to the true ones than are the initial guesses. Using these convergent estimates as the initial values, the algorithm is run on image 1 and image 3, and then image 1 and image 4. The convergent parameters and their corresponding error measure are displayed in Fig. 15(e). The corresponding regions in image 4 are shown in Fig. 15(d). The results are surprisingly good considering that the angle shift between the first and the last image is only 10°, and the data used for each local estimate are only a small region in each image. The ultimate accuracy is obtained by increasing the baseline, and by optimally using the data in all images simultaneously [27].

The algorithm in this paper estimates small surface patches separately. In a complete system, those patches constituting the same primitive surface should be grouped together. Segmentation into primitive surfaces and optimal primitive surface estimation can be achieved using maximum likelihood clustering as developed in [26].

In the above experiment, the gradient descent method requies about ten iterations for each window, and it takes less than one second of running time on a Sun-3 computer. While choosing an alternative surface parameterization and optimizing the codes can reduce the computation time, improvement of the order of more than 10^2 over sequential processing can be achieved with parallel processing because the computation required for each pixel within the window is independent, and can be run in parallel for these several hundred pixels. Therefore, this algorithm is well suited to real-time applications through use of parallel processors.

Most of the surface patches seen in the windows in image 1 are estimated accurately. There are a few windows for which this does not happen, primarily because such a window sees portions of two or more planar surfaces, thus violating the model used which is that each window sees only one surface, or because the surface patch seen in the window is not seen in the second image. However, these mismatches can be detected by checking the size of (14), or by using Bayesian decision theory. If a window in image 1 contains a patch of constant image intensity, then the system knows that a good estimate of 3-D surface patch using only local data cannot be obtained, and therefore does not process that window data here.

Initial experiments with real data are illustrated in Fig. 17(a)–(d). A 242 × 256 pixel CCD camera is mounted on a robot arm. The cube shown in dashed lines in Fig. 17(a) and (d) is the calibration cube with 12″ side, used

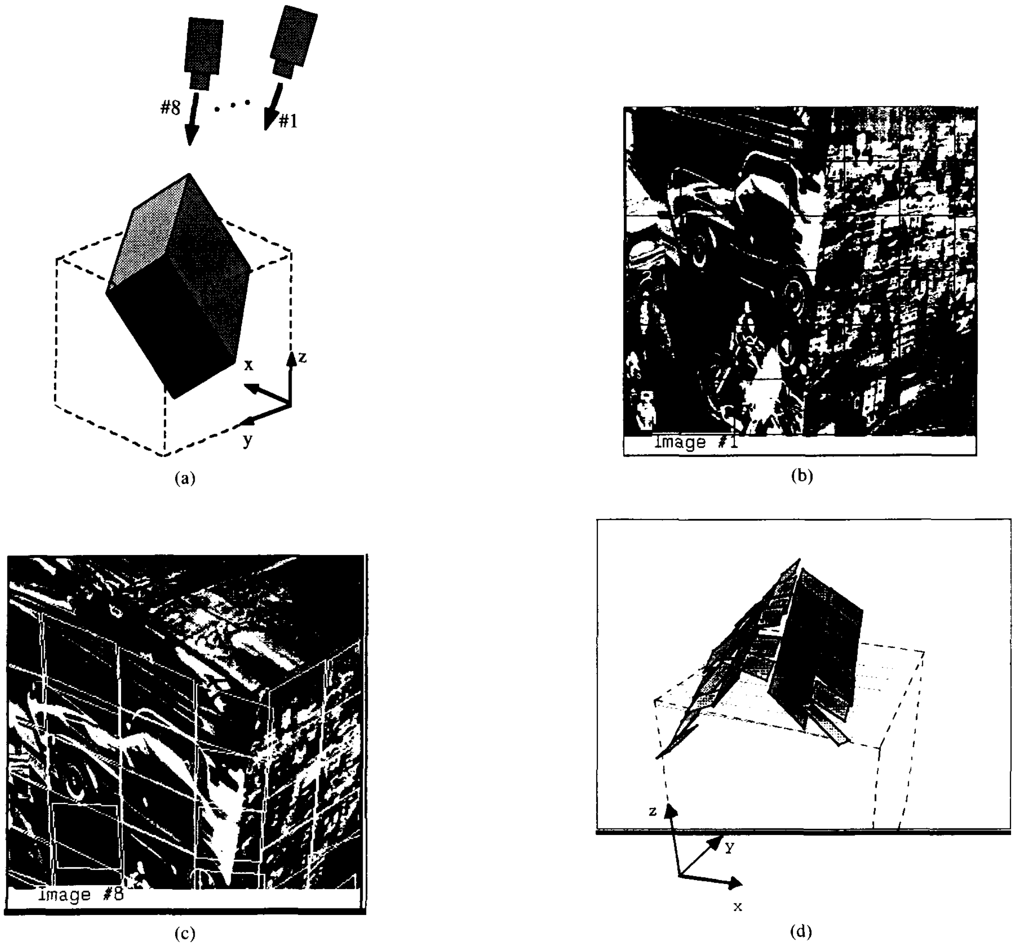


Fig. 17. (a) shows the relative positions between the camera and the object (the shaded rectangular box); (b) and (c) are views of the cube in 1st and 8th camera positions, respectively. (d) shows the computer reconstruction of the 3D surfaces using the estimation results.

for calibrating the camera. It is removed prior to taking the image data. The object to be estimated is a rectangular box, as shown shaded in Fig. 17(a). The box is slightly smaller than the calibration cube. Xeroxed copies of journal pages have been pasted on the box to provide the surface patterns. The camera is moved through a sequence of eight positions. This trajectory with respect to the cube geometry is shown in Fig. 17(a). The camera is roughly 1′ above the box. Angles between the camera optical axes in the 1st and 2nd positions, and the 2nd and 3rd positions are each slightly less than 0.5°. Angles between subsequent pairs of positions are larger, and the angle between the optical axes in the 1st and 8th positions is 14°.

Fig. 17(b) and (c) are views of the cube in 1st and 8th camera positions, respectively. The procedure here is similar to that for Figs. 15 and 16, except that here we are using real data and the pinhole model (perspective projection) for the camera.

Image 1 sees only two surfaces of the cube. The image is divided into 40 × 40 pixel windows. Image 8 shows the windows that correspond to windows in Image 1 based on the algorithm's estimate of the planar surface patch for each window. Notice that most correspondences are good. Since the window in row 4 column 2 in Image 1 consists of essentially constant image intensity, i.e., no pattern, the associated planar surface estimate will be poor, and that can be seen by looking at the estimate of the associated region in Image 8. Also, for some windows in Image 1, the associated region in the camera 8 image plane will be missing some image data, i.e., these windows extend beyond the view angle in the 8th camera position. One of course expects to have poor 3-D surface estimates for these regions. These phenomena can be seen in the estimated 3-D surface patches in Figs. 17(d). Note also that where a window in Image 1 views portions of two surfaces (i.e., on the boundary where two surfaces intersect) the estimated surfaces have orientations between those of the true surfaces.

E. Minimization of $e_D(\boldsymbol{a})$ is Maximum Likelihood Estimation

In this section, we show that the $\boldsymbol{a}$ which minimizes $e_D(\boldsymbol{a})$, (14), is in fact the maximum likelihood estimate (mle) of $\boldsymbol{a}$ for the probabilistic model that we now discuss.

Let $\boldsymbol{r} = (x,y,z)^T$, as before, be the Cartesian coordinates of a point in three-dimensional space. A surface is defined in general as the set of points whose coordinates are functions of two independent parameters. Thus, the equations of a surface can be written as $x = x(t_1, t_2)$, $y = y(t_1, t_2)$, $z = z(t_1, t_2)$ where t_1 and t_2 are the independent parameters. In other words, any point on the surface is uniquely determined by two numbers, t_1 and t_2, and we shall call $\boldsymbol{t} = (t_1, t_2)^T$ the curvilinear coordinates of a point on the surface. Of course, the choice of the curvilinear coordinate system is not unique. Fortunately, we do not have to compute it since our algorithm for estimating $\boldsymbol{a}$ turns out to be independent of the choice of the curvilinear coordinate system.

Let us choose an arbitrary curvilinear coordinate system on that surface which is described by $\boldsymbol{a}$. Let s and $\boldsymbol{u}$ be points in image frames 1 and 2 that are views of points $\boldsymbol{t}$ and $\boldsymbol{t}'$, respectively, on the object surface. Then there are functions $\boldsymbol{q}_1(s, \boldsymbol{a})$ and $\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$ such that

$$\boldsymbol{t} = \boldsymbol{q}_1(s, \boldsymbol{a}), \qquad \boldsymbol{t}' = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}),$$
$$s = \boldsymbol{q}_1^{-1}(\boldsymbol{t}, \boldsymbol{a}), \qquad \boldsymbol{u} = \boldsymbol{q}^{-1}(\boldsymbol{t}', \boldsymbol{b}, \boldsymbol{a}). \tag{24}$$

(The notations $\boldsymbol{q}_1(s, \boldsymbol{a})$ and $\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$ differ because we are taking CRF1 to be the world coordinate system.)

Let $\mu(\boldsymbol{t})$ denote the brightness of the object surface at point $\boldsymbol{t}$. We shall call $\mu(\cdot)$ the surface pattern. The surface pattern $\mu(\boldsymbol{t})$ is seen in image planes 1 and 2 as $\mu_1(s)$ and $\mu_2(\boldsymbol{u})$, respectively. Thus, $\mu_1(s) = \mu(\boldsymbol{q}_1(s, \boldsymbol{a}))$, and $\mu_2(\boldsymbol{u}) = \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}))$. In practice, what is observed at s and $\boldsymbol{u}$ is

$$I_1(s) = \mu_1(s) + w_1(s)$$
$$I_2(\boldsymbol{u}) = \mu_2(\boldsymbol{u}) + w_2(\boldsymbol{u}) \tag{25}$$

where $w_1(s)$, for all s, and $w_2(\boldsymbol{u})$, for all $\boldsymbol{u}$, are zero mean, homogeneous white Gaussian noise processes having common variance σ^2, i.e., they are independent identically distributed (i.i.d.) random variables. Thus, $\mu_1(s) = E[I_1(s)]$ and $\mu_2(\boldsymbol{u}) = E[I_2(\boldsymbol{u})]$ where $E[\cdot]$ denotes expectation of the random variable within the brackets.

Let D_1 and D_2 denote regions in image frames 1 and 2, respectively, and let I_1 and let I_2 be vectors with components $I_1(s)$, $s \in D_1$, and $I_2(\boldsymbol{u})$, $\boldsymbol{u} \in D_2$, respectively. Hence, I_1 and I_2 represent the picture functions over regions D_1 and D_2 in image frames 1 and 2, respectively. Let $\boldsymbol{\mu}$ denote a vector with components $\mu(\boldsymbol{t})$ where the $\boldsymbol{t}$ are points seen in D_1 or D_2 or both. Then the joint likelihood of I_1 and I_2 given $\boldsymbol{a}$, σ^2, and $\mu(\boldsymbol{t})$ for all $\boldsymbol{t}$, is

$$p(I_1, I_2 | \boldsymbol{a}, \mu, \sigma^2)$$
$$= (2\pi\sigma^2)^{-d_1/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{s \in D_1}[I_1(s) - \mu_1(s)]^2\right\}$$
$$\cdot (2\pi\sigma^2)^{-d_2/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{\boldsymbol{u} \in D_2}[I_2(\boldsymbol{u}) - \mu_2(\boldsymbol{u})]^2\right\} \tag{26}$$

where d_1 and d_2 are the numbers of pixels in D_1 and D_2, respectively. We assume here that the pattern on the object surface is an *a priori unknown* nonrandom function, so that $\mu_1(s) = \mu(\boldsymbol{q}_1(s, \boldsymbol{a}))$ and $\mu_2(\boldsymbol{u}) = \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}))$ are *a priori* unknown nonrandom functions. Hence, the joint likelihood (26) depends not only on $\boldsymbol{a}$, but also on σ^2 and $\mu(\boldsymbol{t})$ for all $\boldsymbol{t}$. We seek the maximum likelihood estimate $\hat{\boldsymbol{a}}$ for the unknown $\boldsymbol{a}$. Unfortunately, this requires the simultaneous computation of the maximum likelihood estimates $\hat{\mu}(\boldsymbol{t})$ and $\hat{\sigma}^2$ for the unknowns portions of $\mu(\boldsymbol{t})$ seen in the images, and for σ^2.

The *necessity* for maximum likelihood estimation of $\mu(\boldsymbol{t})$, for those $\boldsymbol{t}$ seen in D_1 or D_2 or both, in order to realize maximum likelihood estimation of $\boldsymbol{a}$, is interesting. In this paper, $\mu(\boldsymbol{t})$ is considered to be completely arbitrary, but in most applications there is some restrictive model for the surface pattern. Some examples of restricted models are one or more parameterized curves, a polynomial intensity function, a locally homogeneous parameterized stochastic process, or some other parameterized model. In all cases, the pattern model parameters must be estimated jointly with the estimation of $\boldsymbol{a}$. Recent examples of this in the literature are [28], [29]. In [28], contours in the image (i.e., curves across which the image intensity changes rapidly) are modeled as polynomial curves with *a priori* unknown coefficients, and the data used with these models are not the original images but rather the edge maps for two or more images. In [29], MRF models are used for textured patterns on *locally planar* 3-D curved surfaces.

For each surface point $\boldsymbol{t}$ seen in image 1 or image 2, the maximum likelihood estimate of $\mu(\boldsymbol{t})$ given $\boldsymbol{a}$ is the $\hat{\mu}(\boldsymbol{t})$ that maximizes (26). Let $\boldsymbol{u} = \boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}_T)$. Then, $\hat{\mu}(\boldsymbol{t})$ is found to be the following.

1) If surface point $\boldsymbol{t}$ is seen at s in D_1 but not in D_2, then $\hat{\mu}(\boldsymbol{t}) = \hat{\mu}_1(s) = I_1(s)$.

2) If surface point $\boldsymbol{t}$ is seen at $\boldsymbol{u}$ in D_2 but not in D_1, then $\hat{\mu}(\boldsymbol{t}) = \hat{\mu}_2(\boldsymbol{u}) = I_2(\boldsymbol{u})$.

3) If surface point $\boldsymbol{t}$ is seen in both frames at $s \in D_1$ and $\boldsymbol{u} \in D_2$, respectively, then

$$\hat{\mu}(\boldsymbol{t}) = \hat{\mu}_1(s) = \hat{\mu}_2(\boldsymbol{u}) = \frac{I_1(s) + I_2(\boldsymbol{u})}{2}.$$

Let D_{12} be a subset of D_1 such that

$$D_{12} = \left\{s\colon s \in D_1, \quad \text{and} \quad \boldsymbol{u} = \boldsymbol{h}\left(s, \boldsymbol{b}, z(s, \boldsymbol{a})\right) \in D_2\right\},$$

i.e., the subset of points in D_1 in image plane 1 that are views of surface points also seen in D_2 in image plane 2. (Note that D_{12} is a function of $\boldsymbol{a}$.) Replacing $\mu(\cdot)$ in (26) by its maximum likelihood estimate $\hat{\mu}(\cdot)$ obtained above, we have

$$p(I_1, I_2 | \boldsymbol{a}, \hat{\mu}, \sigma^2)$$
$$= (2\pi\sigma^2)^{-(d_1+d_2)/2} \cdot \exp\left\{-\frac{1}{2\sigma^2}\right.$$
$$\cdot \sum_{s \in D_{12}}\left[\left(I_1(s) - \frac{I_1(s) + I_2(\boldsymbol{u})}{2}\right)^2\right.$$
$$\left.\left. + \left(I_2(\boldsymbol{u}) - \frac{I_1(s) + I_2(\boldsymbol{u})}{2}\right)^2\right]\right\}$$

$$= (2\pi\sigma^2)^{-(d_1+d_2)/2} \cdot \exp\left\{ -\frac{1}{4\sigma^2} \sum_{s \in D_{12}} [I_1(s) - I_2(\boldsymbol{u})]^2 \right\} \quad (27)$$

where $\boldsymbol{u} = \boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a}))$. Therefore, maximization of the joint likelihood function $p(I_1, I_2 | \boldsymbol{a}, \hat{\mu}, \sigma^2)$ with respect to $\boldsymbol{a}$ is simply minimization of $e_D(\boldsymbol{a})$ in (14) with $D = D_{12}$. Thus, the $\hat{\boldsymbol{a}}$ that minimizes $e_D(\boldsymbol{a})$ is the maximum likelihood estimate of $\boldsymbol{a}$ when the object surface pattern $\mu(\cdot)$ is treated as an *a priori* unknown deterministic function. (As can be seen in (27), it is unnecessary to estimate σ^2 in order to estimate $\boldsymbol{a}$.)

We run into a problem here due to the quantization of the image into pixels. From Fig. 14, we see that the marked region of object surface is seen as two pixels centered at $\boldsymbol{u}$ and $\boldsymbol{u}'$ in image 2, and one pixel centered at s in image 1. More generally, a region on the object surface may be seen as a pixel at s in set D_1 and as pixels at $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n$ in set D_2. In that case, (27) is not completely correct, and a more exact formulation is given in Section II-F. However, in practice the use of (27) for maximum likelihood estimation works well, and we use it.

F. A More Exact Expression for $p(I_1, J_2 | \boldsymbol{a}, \mu, \sigma^2)$

Suppose a region on the object surface is seen as a pixel at s in set D_1 and as pixels at $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n$ in set D_2. Then the contribution to the exponent in (26) of the data at these pixels is

$$-(1/2\sigma^2)\left\{ [I_1(s) - \mu_1(s)]^2 + \sum_{i=1}^{n} [I_2(\boldsymbol{u}_i) - \mu_2(\boldsymbol{u}_i)]^2 \right\} \quad (28)$$

where $\mu_1(s) = (1/n) \Sigma_{i=1}^n \mu_2(\boldsymbol{u}_i)$. This relationship between $\mu_1(s)$ and the $\mu_2(\boldsymbol{u}_i)$ follows from the assumption of Lambertian image formation, i.e., an *incremental* region of the object surface is seen with the same brightness at points in both images. Then the mles for $\mu_1(s)$ and $\mu_2(\boldsymbol{u}_i)$, $i = 1, \cdots, n$, are obtained by maximizing (28) with respect to $\mu_1(s)$ and the $\mu_2(\boldsymbol{u}_i)$ subject to the constraint among these variables. The mles are

$$\hat{\mu}_1(s) = \frac{n}{n+1} \bar{I}_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a})) + \frac{1}{n+1} I_1(s),$$

and

$$\hat{\mu}_2(\boldsymbol{u}_i) = I_2(\boldsymbol{u}_i) - \frac{1}{n+1} \bar{I}_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a})) + \frac{1}{n+1} I_1(s) \quad (29)$$

where $\bar{I}_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}))$ denotes $(1/n) \Sigma_{i=1}^n I_2(\boldsymbol{u}_i)$. Upon replacing $\mu_1(s)$ and the $\mu_2(\boldsymbol{u}_i)$ in (28) by their mles, (28) becomes

$$-(1/2\sigma^2)\left[I_1(s) - \bar{I}_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}))\right]^2. \quad (30)$$

Hence, the contributions of $I_1(s)$ and $I_2(\boldsymbol{u}_i)$, $i = 1, \cdots, n$, to the function that must be maximized in order to obtain the mle of $\boldsymbol{a}$, is (30). Equation (30) is interesting for two reasons. The first is its simplicity and that it is $-(1/2\sigma^2)[I_1(s) - I_2(\boldsymbol{u})]^2$ when $n = 1$. The other reason is that $I_1(s)$ contributes only once to the sum that must be maximized rather than n times as would be the case if $-(1/2\sigma^2) \cdot \Sigma_{i=1}^n [I_1(s) - I_2(\boldsymbol{u})]^2$ were used.

Now, suppose a small region on the object surface is seen as the pixels at $s_1, s_2, \cdots, s_n$ in D_1 and the pixel at $\boldsymbol{u}$ in D_2 where $\boldsymbol{h}(s_i, \boldsymbol{b}, \boldsymbol{a}) \approx \boldsymbol{u}$. Then preceding analogously to the preceding paragraph, we have

$$\hat{\mu}_2(\boldsymbol{u}) = \frac{n}{n+1} \bar{I}_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) + \frac{1}{n+1} I_2(\boldsymbol{u}),$$

and

$$\hat{\mu}_1(s_i) = I_1(s_i) - \frac{1}{n+1} \bar{I}_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) + \frac{1}{n+1} I_2(\boldsymbol{u})$$

where $\bar{I}_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}))$ denotes $(1/n) \Sigma_{i=1}^n I_1(s_i)$. Also, upon replacing $\mu_2(\boldsymbol{u})$ and the $\mu_1(s_i)$ by their mles, the contribution to the exponent of (26) becomes

$$-(1/2\sigma^2)\left[\bar{I}_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) - I_2(\boldsymbol{u})\right]^2 \quad (31)$$

Using results (30) and (31), we see that a more accurate expression for the left side of (27) is

$$p(I_1, I_2 | \boldsymbol{a}, \hat{\mu}, \sigma^2) = (2\pi\sigma^2)^{-(d_1+d_2)/2} \exp\left\{ -\tfrac{1}{2}\sigma^2 \left(\sum_{s \in D'_{12}} \left[I_1(s) - \bar{I}_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}))\right]^2 + \sum_{\boldsymbol{u} \in \boldsymbol{h}(D_{12} - D'_{12}, \boldsymbol{b}, \boldsymbol{a})} \left[\bar{I}_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) - I_2(\boldsymbol{u})\right]^2 \right) \right\} \quad (32)$$

where D'_{12} is the set of points s such that $s \in D_{12}$ and the pixel at s maps onto one or more pixels in D_2. Similarly, $D_{12} - D'_{12}$ is the set of points s belonging to D_{12} and not D'_{12}, and $\boldsymbol{h}(D_{12} - D'_{12}, \boldsymbol{b}, \boldsymbol{a})$ is the set of points $\boldsymbol{u}$ such that the pixel at $\boldsymbol{u}$ corresponds to pixels at two or more s in D_{12}. Now maximization of (32) with respect to $\boldsymbol{a}$ is minimization of the sum of the two summations in the exponent.

There is still a source of inexactness in the use of (32) for representing the left side of (27). It is that one pixel in D_1 will never map onto an integer number of pixels in D_2 and vice versa. One result of this is that even if $\sigma^2 = 0$ and the true value of $\boldsymbol{a}$ is used, the exponent of (32) will not be zero. This effect is equivalent to adding colored noises (i.e., nonwhite noises) to the two images. The standard deviation of the quantization noise at a pixel in image k will be proportional to the gradient of $\mu_k(\cdot)$ at the pixel. When σ^2 is small, this colored noise has a greater effect than does the white noise assumed in this model.

In summary, if $p(I_1, I_2 | \boldsymbol{a}, \hat{\mu}, \sigma^2)$ is to be maximized with respect to $\boldsymbol{a}$, (32) should be used. However, the approximation (27) is computationally simpler and works well.

III. Cramer-Rao Lower Bounds on the Error Covariance for the Estimates of Surface Parameters $\boldsymbol{a}$

In this section, we derive lower bounds for the covariance matrices for the *a priori* unknown parameters of the 3-D objects and the object surface patterns for those situations where the parameters to be estimated are treated as unknown constants or as random variables. Note that these are fundamental bounds that depend on the raw image data and any *a priori* information concerning camera and object surface geometry. No matter what object parameter estimation algorithms are used, the resulting object parameter estimation-errors can never be smaller than these bounds. Furthermore, if the image data windows used contain many pixels and the object surfaces being estimated are smooth, then maximum likelihood parameter estimators will have error covariance matrices given approximately by the C.R. Bound. These bounds require computing the Fisher Information Matrix [13]. We compute this matrix and the error covariance bound. The final results are given in (37) and (41).

Let $\boldsymbol{\alpha}^T = (\boldsymbol{\mu}^T, \sigma^2, \boldsymbol{a}^T)$. Here, $\boldsymbol{\mu}^T$ is a vector having components $\mu(\boldsymbol{t})$ where $\boldsymbol{t}$ is a point in the 2-D parametric space on which the object surface pattern is specified as discussed in Section II-E and (24). The points $\boldsymbol{t}$ in question are those specifying 3-D surface points seen in set D_1 or set D_2 or both, where D_1 and D_2 are arbitrary sets in the image planes in CRF1 and CRF2, respectively. As discussed in Section II-E and (24), through $\boldsymbol{q}_1^{-1}(\boldsymbol{t}, \boldsymbol{a})$ and $\boldsymbol{q}^{-1}(\boldsymbol{t}, \boldsymbol{b}, \boldsymbol{a})$ point $\boldsymbol{t}$ maps to point $s \in D_1$ in image 1 or $\boldsymbol{u} \in D_2$ in image 2, or both. Then the Fisher Information Matrix is[3]

$$F = -E\left[\frac{\partial^2 \ln p(I_1, I_2 \mid \boldsymbol{\alpha}_T)}{(\partial \boldsymbol{\alpha}_T)^2}\right]$$
$$= E\left\{\left[\frac{\partial \ln p(I_1, I_2 \mid \boldsymbol{\alpha}_T)}{\partial \boldsymbol{\alpha}_T}\right]^T \left[\frac{\partial \ln p(I_1, I_2 \mid \boldsymbol{\alpha}_T)}{\partial \boldsymbol{\alpha}_T}\right]\right\} \tag{33}$$

where $\boldsymbol{\alpha}_T$ is the true value of $\boldsymbol{\alpha}$, $\partial \ln p(\cdot, \cdot \mid \boldsymbol{\alpha}_T)/\partial \boldsymbol{\alpha}_T$ is a row gradient vector evaluated at $\boldsymbol{\alpha} = \boldsymbol{\alpha}_T$, and $\partial^2 \ln p(\cdot, \cdot \mid \boldsymbol{\alpha}_T)/(\partial \boldsymbol{\alpha}_T)^2$ is a 2nd partial derivative matrix evaluated at $\boldsymbol{\alpha} = \boldsymbol{\alpha}_T$. Let Λ be the error covariance matrix for *any unbiased* estimator for $\boldsymbol{\alpha}$ based on data in D_1 and D_2. Then the Cramer-Rao bound is

$$F^{-1} \leq \Lambda \tag{33a}$$

where by the inequality we mean that $\Lambda - F^{-1}$ is nonnegative definite. This implies that if $\boldsymbol{v}$ is the parameter estimation error vector, then

$$\text{trace } F^{-1} \leq E[\boldsymbol{v}^T \boldsymbol{v}] = E\left[\sum_i (v_i)^2\right] = \sum_i E[(v_i)^2].$$

Furthermore,

$$ii\text{th element of } F^{-1} \leq E[(v_i)^2],$$

where v_i is the ith element of $\boldsymbol{v}$. Similar results apply when $\boldsymbol{v}$ is a biased estimate of $\boldsymbol{\alpha}_T$ [13]. Let $F_{\boldsymbol{a}}^{-1}$ denote the diagonal subblock of F^{-1} that applies to $\boldsymbol{a}$ alone.

We define $\mu_n(\cdot)$ to be the image mean value function in CRFn. We now explore in detail the properties of the Fisher information matrix, (33). Let $\boldsymbol{u}$ and s be related by $\boldsymbol{u} = \boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a}_T))$. We assume that $\boldsymbol{t} = \boldsymbol{q}_1(s, \boldsymbol{a})$ and $\boldsymbol{t} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$ are 1 : 1 functions on D_1 and D_2, respectively. If they are not 1 : 1, then two or more points on the object surface project onto a single pixel in image plane 1, or onto a single pixel in image plane 2, or this can happen with both image planes. In such a case, we use only one of these surface points in the following development, and that is the surface point that projects closest to the center of the pixel onto which these surface points project. From (24) and using the reasoning leading to (26), we have

$$p(I_1, I_2 \mid \boldsymbol{a}, \mu, \sigma^2)$$
$$= (2\pi\sigma^2)^{-d_1/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{s \in D_1} [I_1(s) - \mu(\boldsymbol{t})]^2\right\}$$
$$\cdot (2\pi\sigma^2)^{-d_2/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{\boldsymbol{u} \in D_2} [I_2(\boldsymbol{u}) - \mu(\boldsymbol{t})]^2\right\}. \tag{26a}$$

If s and $\boldsymbol{u}$ see the same point, specified by $\boldsymbol{t}$, on the object surface, then $\mu_1(s) = \mu(\boldsymbol{t}) = \mu_2(\boldsymbol{u})$.

The exact meaning of (26a) requires some explanation. As explained in Section II-E, the pixel at s in D_1 may see that portion of object surface seen by n pixels $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n$, in D_2. This dependence of the value of $\mu_1(s)$ on the values of $\mu_2(\cdot)$ and $\mu(\cdot)$ at a few values of their arguments poses a complication when trying to find a *simple* form for the dependence of F^{-1} in (33a) on camera geometry, object surface parameterization, and object reflectivity coefficient pattern. (In theory, F^{-1} in (33a) can be computed for the sensor model described.) To arrive at a sensor model that is easier to work with, we assume that the noiseless image gray level at pixel s in D_1 is due to the sensor illumination at s, the center of the pixel, only. Then there is only one of the $\boldsymbol{u}_i'$s, $i = 1, \cdots, n$, for which $\mu_2(\boldsymbol{u}_i) \approx \mu_1(s)$, which is the $\boldsymbol{u}_i$ closest to $\boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a}_T))$. Similarly, there is only one $\mu(\boldsymbol{t}_i)$ for which $\mu(\boldsymbol{t}_i) \approx \mu_1(s)$, namely, the $\boldsymbol{t}_i$, $i = 1, \cdots, n'$, that is closest to $\boldsymbol{q}_1(s, \boldsymbol{a})$. The other $\mu_2(\boldsymbol{u}_i)$ and $\mu(\boldsymbol{t}_i)$ are not constrained to being the same as any of the $\mu_1(s')$, $s' \in D_1$. A similar statement applies when one pixel in D_2 sees a region of object surface seen by n pixels in D_1. This is the model that we now use in this section and Sections III-A, B, and C. (A different model may give slightly different results.)

[3] The following notation is used. For any function $w(\boldsymbol{\alpha})$, by $\partial w(\boldsymbol{\alpha}_T)/\partial \boldsymbol{\alpha}_T$ we mean $\partial w(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}|_{\boldsymbol{\alpha} = \boldsymbol{\alpha}_T}$.

From (24), we have

$$\frac{\partial \mu_1(s)}{\partial a} = \frac{\partial \mu(q_1(s, a))}{\partial a} = \frac{\partial \mu(t)}{\partial t}\frac{\partial q_1(s, a)}{\partial a}\bigg|_{t=q_1(s,a)}$$

$$\frac{\partial \mu_2(u)}{\partial a} = \frac{\partial \mu(q(s, b, a))}{\partial a} = \frac{\partial \mu(t)}{\partial t}\frac{\partial q(u, b, a)}{\partial a}\bigg|_{t=q(u,b,a)}. \qquad (34)$$

Note, in the above, $\partial q_1(s, a)/\partial a$ is a $2 \times K$ matrix where K is the number of components in a. Similarly for $\partial q(u, b, a)/\partial a$.

Define $\tilde{D}_{12}$ as the set of s such that $s \in D_1$ and s belongs to one of the following sets: the pixel at s maps onto one or more pixels in D_2; or two or more pixels in D_1 map onto the pixel at u in D_2, and, of these pixel centers, s is the one for which $h(s, b, z(s, a))$ is the closest to u.

From (26a) and (33), following some substantial work, we arrive at (details are in Appendix B1):

$$F_a^{-1} = \frac{\sigma_T^2}{2}\left\{\sum_{s\in\tilde{D}_{12}}\left[\frac{\partial \mu(q_1(s, a_T))}{\partial a_T} - \frac{\partial \mu(q(u, b, a_T))}{\partial a_T}\right]^T \cdot \left[\frac{\partial \mu(q_1(s, a_T))}{\partial a_T} - \frac{\partial \mu(q(u, b_2, a_T))}{\partial a_T}\right]\right\}^{-1}. \qquad (35)$$

Note that due to cancellations, only the subset $\tilde{D}_{12}$ of D_1 contributes to the final expression for the bound (35). This is as expected since data points cannot contribute to the estimation of a unless there is a point in each image that is due to the same point on the object surface. The right side of (35), the Cramer-Rao Bound for the estimation error for the parameter vector a_T, is interesting. Note that the Cramer-Rao (C.R.) Bound is inversely dependent on th differences in the gradient vectors $\partial\mu_1(s)/\partial a_T = \partial\mu(q_1(s, a_T))/\partial a_T$ and $\partial\mu_2(u)/\partial a_T = \partial\mu(q(u, b, a_T))/\partial a_T$, i.e., the gradients with respect to a of the image mean values taken by cameras 1 and 2, respectively. Equivalently, upon using (34), this difference can be written as

$$\left[\frac{\partial \mu_1(s)}{\partial a_T} - \frac{\partial \mu_2(u)}{\partial a_T}\right] = \frac{\partial \mu(t)}{\partial t} \cdot \left[\frac{\partial q_1(s, a_T)}{\partial a_T} - \frac{\partial q(u, b, a_T)}{\partial a_T}\right]. \qquad (36)$$

Hence, in order to have a small C.R. bound, one wants the difference in the rate of change of t with respect to a_T in CRF1 and CRF2 to be as large as possible, i.e., one wants the columns of the matrix in the brackets in (36) to be as large as possible.

A. Cramer-Rao Lower Bounds when Using an Alternative Pattern Parameterization

It is often convenient to take the *a priori* unknown pattern to be $\mu_1(s)$, the mean value function for image 1, or $\mu_2(u)$ rather than $\mu(t)$. There are two reasons for this. First, since the data sets that we are dealing with are I_1 and I_2, the easiest pattern parameterization to work with is often the mean value function associated with one of these sets. Second, though in theory we can always specify an object surface pattern as a 2-D parameterization on the object surface, in practice this may be messy. From a physical point of view there is a slight difference between assuming the true fixed pattern is $\mu(t)$ and assuming it is $\mu_1(s)$. If one lets a vary, then in the first case the mean value function for the first image varies at point s because it is given by $\mu(q_1(s, a))$, i.e., because the object surface moves. However, in the second case $\mu_1(s)$ is fixed and does not vary with a, but the pattern at a point on the object surface will vary as a varies. In general, if $\mu_1(s)$ is taken to be the *a priori* unknown pattern parameters, then some of the elements in the resulting Fisher Information Matrix will differ from those in (b10). However, an interesting result easily proven using the types of identities developed in Appendix B2, is that the C.R. Bound for the a vector will be the same whether the unknown pattern vector is taken to be $\mu_1(s)$, or $\mu_2(u)$, or $\mu(t)$. Since the theory is a little easier to interpret for $\mu_1(s)$ [or $\mu_2(u)$] as the unknown pattern parameters, the remainder of our development of the C.R. Bound is for this case.

Hence, let $\mu_1(s)$, $s \in D_1$, be the *a priori* unknown pattern parameters. Then in Appendix B2 it is argued that the C.R. Bound for a_T, F_a^{-1}, is given by

$$2\sigma_T^2\left\{\sum_{s\in\tilde{D}_{12}}\left[\frac{\partial \mu_2(h(s, b, a_T))}{\partial a_T}\right]^T\left[\frac{\partial \mu_2(h(s, b, a_T))}{\partial a_T}\right]\right\}^{-1} \qquad (37)$$

where $\tilde{D}_{12}$ is defined before (35). (See [17] for details.) Note that the true unknown pattern parameters are assumed to be the $\mu_1(s)$ for all s, but it is convenient to express the bound in terms of $\mu_2(u)$ where $\mu_2(h(s, b, a_T)) = \mu_1(s)$.

To explore (37) further, observe that it can be expressed as

$$2\sigma_T^2\left\{\sum_{s\in\tilde{D}_{12}}\left(\frac{\partial h(s, b, z(s, a_T))}{\partial a_T}\right)^T \cdot \left[\left(\frac{\partial \mu_2(u)}{\partial u}\right)^T\left(\frac{\partial \mu_2(u)}{\partial u}\right)\right] \cdot \left(\frac{\partial h(s, b, z(s, a_T))}{\partial a_T}\right)\right\}^{-1}_{u=h(s,b,z(s,a_T))}. \qquad (38)$$

From (16),

$$\frac{\partial h(s, b, z(s, a_T))}{\partial a_T} = c_{12}\frac{\partial z(s, a_T)}{\partial a_T}, \qquad (39)$$

so that

$$\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\Bigg|_{\boldsymbol{u}=\boldsymbol{h}(s,\boldsymbol{b},z(s,\boldsymbol{a}_T))} = \left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \boldsymbol{c}_{12}\right] \frac{\partial z(s, \boldsymbol{a}_T)}{\partial \boldsymbol{a}_T}. \tag{40}$$

Then (38) is

$$F_a^{-1} = 2\sigma_T^2 \left\{ \sum_{s\in\tilde{D}_{12}} \left[\left(\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \boldsymbol{c}_{12}\right)\left(\frac{\partial z(s, \boldsymbol{a}_T)}{\partial \boldsymbol{a}_T}\right)^T \cdot \left(\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \boldsymbol{c}_{12}\right)\left(\frac{\partial z(s, \boldsymbol{a}_T)}{\partial \boldsymbol{a}_T}\right)\right]\right\}^{-1} \tag{41}$$

Expressions for $\partial z(s, \boldsymbol{a})/\partial \boldsymbol{a}$ are derived and given in Section II-A for the sphere and Appendixes C, D, and E for the plane, cylinder, and general quadric, respectively. By assuming a deterministic or a stochastic model for $\mu_1(s)$, it is sometimes possible to obtain a closed form expression for (41). An example of this is given in Section III-D.

B. Interpretation of the Cramer-Rao Bound

The C.R. Bound (41) has a very simple, physically meaningful interpretation. First, observe that the dependence of the bound on each of camera geometry, object surface pattern, and object surface shape is immediately seen. Specifically, the two-component vector $\boldsymbol{c}_{12}$ is the projection in CRF2 of a unit vector in CRF1 at the origin in the direction of the z-axis. It represents the influence of the two-camera geometry on the Bound. The vector $\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{u}$ is the gradient of the mean value of image 2 intensity function. It represents the contribution of the object surface pattern to the bound. Finally, the vector $\partial z(s, \boldsymbol{a})/\partial \boldsymbol{a}$ is the contribution of the object surface shape to the Bound. It represents the dependence of object surface shape on the parameter vector $\boldsymbol{a}$.

We develop the interpretation of (41) and (38) further. Note that because of the use of the orthographic projection, $\boldsymbol{c}_{12}$ gives the direction of all epipolar lines in CRF2. (By an epipolar line we mean the following. Each point in image 1 is the image of a point on a 3-D surface. The ray that goes through a 3-D surface point and its image point in image 1 is seen as the so-called epipolar line in the image plane of camera 2. Hence, the point in image 2 that is the image of the 3-D surface point, must lie on this epipolar line.) The magnitude of $\boldsymbol{c}_{12}$ varies as the sine of the angle between the optical axes of the two cameras. We see that this will be very small for optical axes that are almost parallel, and is a maximum for optical axes that are orthogonal.

The partial derivative $\partial z(s, \boldsymbol{a})/\partial a_i$ is the rate of change with respect to parameter a_i of the z component of the 3-D surface point (s^T, z) in CRF1. (It is the rate of change with respect to a_i of distance to the object surface at point (s^T, z).) Hence $|\boldsymbol{c}_{12}\partial z(s, \boldsymbol{a})/\partial a_i|$ is the magnitude of a directional derivative—the rate of change with respect to a_i of the image in the camera2 image plane of the point at $(s^T, z(s, \boldsymbol{a}))$ on the 3-D object surface.

Since $\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{u}$ is the gradient of image 2 intensity, we see that $[\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{u}]\boldsymbol{c}_{12}[\partial z(s, \boldsymbol{a})/\partial a_i]$ is the rate of change with respect to a_i of the intensity in image 2 in the direction of the vector $\boldsymbol{c}_{12}$ at point $\boldsymbol{u} = \boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a}))$. Because of the inverse operation in (41), we make the qualitative statement that the larger these directional derivatives are, the smaller will be the covariance matrix for the estimation error for $\boldsymbol{a}_T$.

From the preceding, it is clear that maximum parameter estimation accuracy is obtained for the following conditions. The mean value function, $\mu_2(\boldsymbol{u})$, of the image of the object surface pattern should be rapidly varying so that $\partial \boldsymbol{\mu}_2(\boldsymbol{u})/\partial \boldsymbol{u}$ is large. The angles between the optical axes should be large (90° is the best) in order that $\boldsymbol{c}_{12}$ be large, and the angle between the image intensity gradient and the direction of the epipolar lines should be small. Then $\boldsymbol{c}_{12}[\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{u}]$ will be large. Last, it is desirable that the gradients $\partial z(s, \boldsymbol{a}_T)/\partial \boldsymbol{a}_T$ be large and that their directions be distributed over the entire space as s varies throughout $\tilde{D}_{12}$. In general, these conditions are increasingly better achieved with increasing $\tilde{D}_{12}$, i.e., increasing patch size. If there is flexibility to choose the direction for at least one camera axis during the surface estimation procedure, the axis should be chosen such that $[\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{u}]\boldsymbol{c}_{12}$ is large for most $\boldsymbol{u}$ in $\boldsymbol{h}(\tilde{D}_{12}, \boldsymbol{b}, \boldsymbol{a})$. The choice can be made more specific depending on the situation encountered.

C. Another Interpretation Associated with the Fisher Information Matrix

Recall that

$$e_{\tilde{D}_{12}}(\boldsymbol{a}) = \sum_{s\in\tilde{D}_{12}} \left[I_1(s) - I_2(\boldsymbol{h}(s, \boldsymbol{b}, z(s, \boldsymbol{a})))\right]^2. \tag{14}$$

Assume that the additive noise is 0, so that $I_1(s) = \mu_1(s)$ and $I_2(\boldsymbol{u}) = \mu_2(\boldsymbol{u})$. Expanding the resulting function $e'_{\tilde{D}_{12}}(\boldsymbol{a})$ about the point $\boldsymbol{a}_T$ in a Taylor series up through quadric terms gives us

$$e'_{\tilde{D}_{12}}(\boldsymbol{a}) \approx e'_{\tilde{D}_{12}}(\boldsymbol{a}_T) + \left(\frac{\partial e'_{\tilde{D}_{12}}(\boldsymbol{a}_T)}{\partial \boldsymbol{a}_T}\right)(\boldsymbol{a} - \boldsymbol{a}_T) + \frac{1}{2}(\boldsymbol{a} - \boldsymbol{a}_T)^T\left(\frac{\partial^2 e'_{\tilde{D}_{12}}(\boldsymbol{a}_T)}{(\partial \boldsymbol{a}_T)^2}\right)(\boldsymbol{a} - \boldsymbol{a}_T). \tag{42}$$

But $e'_{\tilde{D}_{12}}(\boldsymbol{a}_T) = 0$, and $\partial e'_{\tilde{D}_{12}}(\boldsymbol{a}_T)/\partial \boldsymbol{a}_T = 0$. Furthermore,

$$\frac{\partial^2 e'_{\tilde{D}_{12}}(\boldsymbol{a}_T)}{(\partial \boldsymbol{a}_T)^2} = \sum_{s\in\tilde{D}_{12}} 2\left(\frac{\partial \mu_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right)^T \cdot \left(\frac{\partial \mu_2(\boldsymbol{h}(s, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right). \tag{43}$$

Hence,

$$e'_{\tilde{D}_{12}}(\boldsymbol{a}) \approx \sum_{s \in \tilde{D}_{12}} (\boldsymbol{a} - \boldsymbol{a}_T)^T \left[\left(\frac{\partial \mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T} \right)^T \cdot \left(\frac{\partial \mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T} \right) \right] (\boldsymbol{a} - \boldsymbol{a}_T). \tag{44}$$

Recall from (37) that the inverse of (43) multiplied by σ_T^2 is the C.R. Bound for the achievable error in estimating $\boldsymbol{a}_T$. From (44), we see that (43) also determines $e'_{\tilde{D}_{12}}(\boldsymbol{a})$. The function $e'_{\tilde{D}_{12}}(\boldsymbol{a})$ is narrow if and only if (43) is large. Hence, we see that the C.R. Bound is small if and only if $e'_{\tilde{D}_{12}}(\boldsymbol{a})$ is a narrow function.

D. An Example of a Numerical Computation of the Cramer-Rao Bound

Some additional insight is provided by a simple example for (41). Let the 3-D surface be the plane

$$z = \rho_0 + \rho_1 s_1 + \rho_2 s_2$$

and let $\mu_2(u)$ be linear,

$$\mu_2(\boldsymbol{u}) = \gamma_0 + \gamma_1 u_1 + \gamma_2 u_2$$

where $(s_1, s_2)^T = \boldsymbol{s}$ and $(u_1, u_2)^T = \boldsymbol{u}$. (This choice for $\mu_2(\boldsymbol{u})$ means that $\mu_1(\boldsymbol{s})$ will also be linear.) Then $\partial\mu_2(\boldsymbol{u})/\partial\boldsymbol{u} = (\gamma_1, \gamma_2)$, so that $[\partial\mu_2(\boldsymbol{u})/\partial\boldsymbol{u}]\boldsymbol{c}_{12}$ is a constant independent of $\boldsymbol{u}$, or, equivalently, of $\boldsymbol{s}$ where $\boldsymbol{u} = \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, z(\boldsymbol{s}, \boldsymbol{a}_T))$; it can therefore be taken out of the summation in (41). Since $\partial z(\boldsymbol{s}, \boldsymbol{a}_T)/\partial\boldsymbol{a}_T = (1, s_1, s_2)$, we see that (41) is

$$2\sigma_T^2 \left\{ \left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \boldsymbol{c}_{12} \right]^2 \begin{bmatrix} N^2 & \frac{1}{2}(N-1)N^2 & \frac{1}{2}(N-1)N^2 \\ \frac{1}{2}(N-1)N^2 & \frac{1}{3}\quad N - \frac{1}{2}\quad (N-1)N^2 & \frac{1}{4}(N-1)^2N^2 \\ \frac{1}{2}(N-1)N^2 & \frac{1}{4}(N-1)^2N^2 & \frac{1}{3}\quad N - \frac{1}{2}\quad (N-1)N^2 \end{bmatrix} \right\}^{-1}$$

where we have used $\Sigma_{i=0}^{N-1}\ i = (1/2)(N-1)N$ and $\Sigma_{i=0}^{N-1}\ i^2 = (1/3)(N-1)(N-(1/2))N$. Upon carrying out the matrix inversion and then keeping only the largest power of N, we have the approximation

$$\left[\frac{2\sigma_T^2}{\left(\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} \boldsymbol{c}_{12} \right)^2} \right] \begin{bmatrix} 7N^{-2} & -6N^{-2} & -6N^{-3} \\ -6N^{-3} & 12N^{-4} & 0 \\ -6N^{-3} & 0 & 12N^{-4} \end{bmatrix} \tag{41a}$$

for the C.R. Bound for $E[(\hat{\boldsymbol{a}} - \boldsymbol{a}_T)(\hat{\boldsymbol{a}} - \boldsymbol{a}_T)^T]$. This becomes the exact bound as N becomes large. The diagonal elements of (41a), starting with the top, are the variances of the errors in estimating ρ_0, ρ_1, and ρ_2, respectively. We see that the variance of ρ_0, the position of the 3-D plane, is inversely proportion to image patch size, the number of pixels used in the stereo estimation. But the variance of the error in estimating plane orientation decreases as the square of this rate, because the further away the image pixels are from the patch center, the more they contribute to the plane orientation estimation accuracy. We feel that the greatest value of the expressions for the C.R. Bound is a feeling for how the parameter estimation error depends on the object surface pattern and the camera and object geometry. However, it may also be of interest to look at numerical examples. Hence, consider $|\partial\mu_2(\boldsymbol{u})/\partial\boldsymbol{u}| = 3$, $\sigma_T^2 = 4$, angle between $(\partial\mu_2(\boldsymbol{u})/\partial\boldsymbol{u})^T$ and $\boldsymbol{c}_{12}$ equal 45°, and $|\boldsymbol{c}_{12}| = 0.17$ (corresponding to an angle of 10° between the optical axes of the two cameras). Note that σ_T is due to the additive noise in our models, but in practice it can account for quantization and other numerical errors in the entire measurement system. Then for $N = 16$ pixels, we have $6.5 \leq \mathrm{Var}\,\rho_0$, and $0.043 \leq \mathrm{Var}\,\rho_1$ and $\mathrm{Var}\,\rho_2$. Hence, the lower bounds for the standard deviations of the slopes of the plane in the vertical or the horizontal directions are 0.21.

IV. Conclusions

A parametric modeling and statistical estimation approach has been proposed and simulations shown for estimating 3-D object surfaces from images taken by calibrated cameras in two positions. The parameter estimation suggested is gradient descent, though other search strategies are also possible. Processing image data in blocks (windows) is central to our approach. The estimation is estimation of patches of 3-D surface by searching in parameter space to simultaneously determine and use the appropriate pair of image regions, one from each image, and to use these for estimating a 3-D surface patch. Though the choice of performance functional was motivated by consideration of engineering reasonableness, we derive the expression for the joint likelihood of the two images, and show that the algorithm is a maximum likelihood parameter estimator, and thus enjoys the desirable estimation accuracy properties of maximum likelihood estimators. A very important concept arising in the maximum likelihood estimation of 3-D surfaces is that the patterns on the 3-D surfaces must also be modeled and estimated. We do this for the case of completely arbitrary patterns, in this paper, and deal with restricted pattern classes elsewhere [28]. Finally, Cramer-Rao Lower Bounds are derived for the covariance matrices for the errors in estimating the *a priori* unknown object surface shape parameters. No surface reconstruction algorithm can be more accurate than these bounds, but the accuracy of our maximum likelihood estimator approximates these bounds as the size of the image patches used becomes large.

Following are a few points that may answer some questions occurring to a reader.

1) In Appendix A, we derive the orthographic projection approximation to the more exact perspective projection. It can be made to be arbitrarily accurate by applying it to suitably small 3-D regions—a different approximation for each such region. Our primary purpose in using this approximation is that a C.R. Bound for the perspective projection would probably be uninterpretable, and the C.R. Bound based on the orthographic projection model probably captures the behavior of the C.R. Bound based on the perspective projection very well.

2) Though the surface estimation algorithm in this paper uses the orthographic projection model, with trivial changes the approach applies to the perspective projection model. However, with the perspective projection, the transformation from image 1 to image 2 is such that there is no longer a computational advantage to computing the gradient (15) directly. Rather, we compute the gradient by computing directional derivatives by evaluating (14) at the appropriate points. The estimation accuracy is good, but the required computation becomes a few times larger than is the computation based on the orthographic projection. The perspective projection model is used in the surface reconstruction experiment in Fig. 17 where the data used are a sequence of real images taken by a CCD camera mounted on a moving robot arm.

3) A major use of the C.R. Bound is an understanding of the relative importance of various factors on accuracy. The way image gradient, camera geometry, and 3-D surface parameter dependence enter (37) appears to be very fundamental and should prove useful even when the true model differs somewhat from the assumed one. However, it is possible to derive exact C.R. Bounds for other models. For example, if the images have low noise, image quantization into pixels can introduce a sizable noise. If at a point the image intensity varies by Q units over an interval equal to the extent of a pixel, then the image quantized into pixels can have an equivalent noise fluctuation of $Q/2$ at the pixel in question. This noise is image dependent and is usually correlated. A C.R. Bound can be derived for this case.

4) Accurate 3-D surface reconstruction requires the use of dependency among points within sizable regions of the surface. An effective way of imposing such dependence is by using prior knowledge of 3-D surface structure. Since planes, spheres, and cylinders are effective in efficiently modeling most man-made objects [30], we have emphasized them. The more *a priori* information one uses in surface reconstruction (or for that matter in any inferencing problem), the more accurate will be the result. And maximum accuracy is obtained by using all of this *a priori* information at the time of processing the raw data.

5) We do not have the entire system sufficiently well calibrated in order to determine the exact surface reconstruction accuracy based on real data. One reason for running the algorithm on the semi-artificial data used in the experiments in the paper, is that we can report this accuracy exactly. The images used in our experiments had low sensing noise, so the existing noise was largely colored image-dependent noise due to pixel quantization. As seen, our algorithms work well in the presence of this noise. Moreover, recent experiments (e.g., Fig. 17) indicate that the reconstruction works very well with real data too. From these experimental results, we feel that this algorithm is robust to deviations from the assumed model that may be encountered.

6) Our performance functionals can be multimodal. Some feeling for the width of the main lobe is given by the curves shown for the experiments reported on in the paper. The true value of the surface parameter vector $\boldsymbol{\alpha}_T$ is always at the global minimum of the performance functional (14) if the image noise is 0. If the noise is nonzero, the $\hat{\boldsymbol{\alpha}}$ at the global minimum of (14) converges to $\boldsymbol{\alpha}_T$, under weak conditions, as the size of the image patches used becomes large. In statistics jargon, the mle is consistent. However, the performance functional can be almost flat in the vicinity of its global minimum. The shape of the functional there depends on the image intensity, surface geometry, and camera positions. The interpretation in Sections III-B and III-C of the C.R. Bound provides insight into what determines how flat or steep the performance functional is. One of the factors that greatly affects the width of the main valley of the performance functional is the angle between the optical axes of the two cameras. The width decreases with increasing angle. This property is exploited to provide our computationally simple algorithm of Section II-D. We begin with a small angle, so that we can use an arbitrary first guess for the surface parameter vector, and then minimize the unimodal performance function. The resulting estimate of $\boldsymbol{\alpha}_T$ is highly inaccurate, but it is accurate enough to lie in the global valley of the performance functional resulting from use of a larger angle. Then, starting with the $\boldsymbol{\alpha}_T$ estimate found, $\boldsymbol{\alpha}_T$ is reestimated with this larger angle. This approach is repeated a few times until an accurate estimate of $\boldsymbol{\alpha}_T$ is obtained.

7) Using our approach, we can do maximum likelihood *pointwise* estimation of 3-D surfces, but we do it by assuming a surface patch model. Then the mle of a point on the surface patch is the point on the estimated surface patch that corresponds to the image of the point. If the image point is $\boldsymbol{s}$ in image 1, then the estimated 3-D point is $(\boldsymbol{s}^T, z)^T$ where z is the value satisfying $g(z, \boldsymbol{s}, \hat{\boldsymbol{\alpha}}) = 0$. Since mle's have minimum variance as data patch size becomes large under fairly general conditions, our estimators should pretty much have maximum accuracy. If the object surface has special modelable pattern structure such as contours, across which the patterns are discontinuous and along which they are smooth, then greatest accuracy is achieved by using this information. We do this in [28].

8) Insights into appropriate image and surface patch sizes to use can be provided in terms of image noise, surface curvature, etc., but this would require some lengthy

development. In brief, if the image noise is white, then by using concepts and techniques such as those in [12], [26], it is possible to determine approximately the error covariance matix for estimating the parameter vector for a surface patch, and through a combination of analysis and experimentation, it is possible to estimate the probability of correctly recognizing which of a number of 3-D surface models is correct for a certain surface patch. The C.R. Bound (41a) also provides insights on estimation accuracy. On a more informal level, an image patch size should be chosen to be large when the standard deviation of noise (either from measurement or from quantization) is large. The more parameters that must be estimated, for a surface patch, the larger should be the data patches that are used. For example, in the presence of modest noise, small planar or spherical patches can be estimated that fit the sphere surface well, but the estimate of the sphere radius and center may be greatly in error. In order to estimate these parameters well, it may be necessary to use a patch covering one eighth or more of the sphere surface. An unconstrained quadric patch has even more parameters to be estimated. However, accurate estimates of curvature can be obtained with small patches if a sequence of images is used [27].

9) We see two ways for estimating large complex surfaces from small patches. One approach is to model a surface as a stochastic process. This is done in [3] where the model is a continuous surface of planar patches with the set of parameter vector values, one vector value for each patch, modeled as a Markov random field (MRF). This MRF provides the *a priori* distribution (knowledge), for the global surface, and indirectly contains information such as surface curvature, blob sizes, etc. There it is shown how the estimators of the present paper fit into the more global scheme. The other approach is to use the maximum likelihood clustering approach of [26] to first estimate small surface patches, and then cluster these into large regions with each large region associated with a single surface model, e.g., with a single smooth surface free of gradient discontinuities.

As we point out, because of the probabilistic formulation of the problem, the powerful machinery of Bayesian inference can be brought to bear in our approach. Included here is approximately Bayesian recognition of the object surface shape class associated with the block of data under consideration. That is, recognition of which of a sphere, a cylinder, a plane, some other parameterized surface, or two or more surfaces are associated with the data block. (The asymptotic Bayesian recognition methods in [12, Section V] are directly applicable here.) Also, the approaches in the preceding point, 9, are Bayesian.

Appendix A
Orthographic Approximation to Pinhole Camera Model

Suppose the ORF is chosen to be the same as CRF1, and the 2-D image coordinates are in the same units as are the 3-D coordinates. Let P be an arbitrary surface point seen in image 1. Let $\boldsymbol{r} = (x\ y\ z)^T$ be the 3-D coordinates of P with respect to the ORF, and $\boldsymbol{s} = (s_x\ s_y)^T$ be its corresponding 2-D image coordinates in image 1. Then, upon using the homogeneous coordinate transformation,

$$\begin{bmatrix} s_x \cdot w \\ s_y \cdot w \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

That is,

$$\boldsymbol{s} = \left\{ \begin{array}{l} s_x = \dfrac{f \cdot x}{f - z} = \eta_x(\boldsymbol{r}) \\ s_y = \dfrac{f \cdot y}{f - z} = \eta_y(\boldsymbol{r}) \end{array} \right\} = \boldsymbol{\eta}(\boldsymbol{r})$$

where f is the camera focal length. Let $\boldsymbol{r}_0 = (x_0\ y_0\ z_0)^T$ denote a 3-D point seen in the image. A Taylor series expansion about $\boldsymbol{r}_0$ is

$$\boldsymbol{s} = \boldsymbol{\eta}(\boldsymbol{r}) = \boldsymbol{\eta}(\boldsymbol{r}_0) + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{r}}(\boldsymbol{r}_0) \cdot (\boldsymbol{r} - \boldsymbol{r}_0) + \cdots$$

$$= \begin{bmatrix} \dfrac{f \cdot x_0}{f - z_0} \\ \dfrac{f \cdot y_0}{f - z_0} \end{bmatrix} + \begin{bmatrix} \dfrac{f}{f - z_0} & 0 & \dfrac{f \cdot x_0}{(f - z_0)^2} \\ 0 & \dfrac{f}{f - z_0} & \dfrac{f \cdot y_0}{(f - z_0)^2} \end{bmatrix} \cdot \begin{bmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{bmatrix} + \cdots .$$

Therefore, we can approximate $\boldsymbol{s}$ as a linear function of x, y, z

$$s_x \approx \frac{f}{f - z_0}\left[x + x_0 \frac{(z - z_0)}{f - z_0}\right]$$

$$s_y \approx \frac{f}{f - z_0}\left[y + y_0 \frac{(z - z_0)}{f - z_0}\right]. \qquad \text{(a1)}$$

This approximation is quite accurate for $|z - z_0|/|f - z_0|$ much smaller than 1. The optical axis of the camera is the ray $x = y = 0$. Suppose (x, y, z) is a point on an object surface, and let x_0, y_0 be the center of an image window.

If the object-to-camera distance is at least a few focal lengths, i.e., $|z_0/f| >> 1$; the object diameter to object-to-camera distance is small, i.e., $|z - z_0|/|f - z_0| <<$ 1 such that $|x_0(z - z_0|/|f - z_0)| << |x|$ and $|y_0(z - z_0)/(f - z_0)| << |y|$; then (a2) is a good approximation to (a1) within the window.

$$s_x \approx \frac{f}{f - z_0} x \qquad s_y \approx \frac{f}{f - z_0} y. \qquad \text{(a2)}$$

If this s_x and this s_y are normalized by dividing by the known scale factor $f/(f - z_0)$, the resulting coordinates of the image point are the 3-D point coordinates x and y, respectively. This is the orthographic imaging model used in the body of the paper.

Appendix B1
Derivation of the C.R. Bound

We briefly derive the C.R. Bound for $\boldsymbol{a}_T$. We begin with (26a).

$$\frac{\partial \ln p(I_1, I_2 | \alpha)}{\partial \mu(\boldsymbol{t})} = \frac{1}{\sigma^2}\left\{[I_1(\boldsymbol{s}) - \mu(\boldsymbol{t})]\delta_1(\boldsymbol{t}) + [I_2(\boldsymbol{u}) - \mu(\boldsymbol{t})]\delta_2(\boldsymbol{t})\right\} \quad \text{(b1)}$$

where $\delta_k(\boldsymbol{t})$ takes values 1 or 0 according to whether or not the point $\boldsymbol{t}$ is seen at a pixel in D_k. Hence, $\delta_1(\boldsymbol{t}) = 1$ for all $\boldsymbol{t} = \boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a})$, $\boldsymbol{s} \in D_1$, and $\delta_2(\boldsymbol{t}) = 1$ for all $\boldsymbol{t}$ such that $\boldsymbol{t} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$, $\boldsymbol{u} \in D_2$. Furthermore, suppose the pixel at $\boldsymbol{s}$ in D_1 maps onto two or more pixels centered at points $\boldsymbol{u}$ in D_2. Then we will consider that pixel center $\boldsymbol{u}$ that is closest to $\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a})$ to be such that $\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{t} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$, and for this $\boldsymbol{t}$, $\delta_1(\boldsymbol{t}) = \delta_2(\boldsymbol{t}) = 1$. A similar statement applies if the pixel at $\boldsymbol{u}$ in D_2 maps onto two or more pixels centered at $\boldsymbol{s}$ in D_1. From (b1),

$$\hat{\mu}(\boldsymbol{t}) = [I_1(\boldsymbol{s})\,\delta_1(\boldsymbol{t}) + I_2(\boldsymbol{u})\,\delta_2(\boldsymbol{t})]/[\delta_1(\boldsymbol{t}) + \delta_2(\boldsymbol{t})]. \quad \text{(b2)}$$

From (b2) it is seen that $\hat{\mu}(\boldsymbol{t})$ is an unbiased estimate. Continuing, we obtain

$$\frac{\partial \ln p(I_1, I_2 | \boldsymbol{\alpha})}{\partial \sigma^2} = -\frac{d_1 + d_2}{2\sigma^2} + \frac{1}{2\sigma^4}\left\{\sum_{\boldsymbol{s} \in D_1} [I_1(\boldsymbol{s}) - \mu_1(\boldsymbol{s})]^2 + \sum_{\boldsymbol{u} \in D_2} [I_2(\boldsymbol{u}) - \mu_2(\boldsymbol{u})]^2\right\}.$$

Hence,

$$\hat{\sigma}^2 = \frac{1}{d_1 + d_2}\left\{\sum_{\boldsymbol{s} \in D_1} [I_1(\boldsymbol{s}) - \hat{\mu}_1(\boldsymbol{s})]^2 + \sum_{\boldsymbol{u} \in D_2} [I_2(\boldsymbol{u}) - \hat{\mu}_2(\boldsymbol{u})]^2\right\} \quad \text{(b3)}$$

where $\hat{\mu}_1(\boldsymbol{s}) = \hat{\mu}(\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a}))$ and $\hat{\mu}_2(\boldsymbol{u}) = \hat{\mu}(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}))$. From (b3), we see that $\hat{\sigma}^2$ is asymptotically unbiased as D_1 and D_2 become large if and only if $D_1 \approx D_{12}$ where D_{12} is defined in Section II-E.

$$\frac{\partial \ln p(I_1, I_2 | \boldsymbol{a})}{\partial \boldsymbol{a}} = \frac{1}{\sigma^2}\left\{\sum_{\boldsymbol{s} \in D_1} [I_1(\boldsymbol{s}) - \mu_1(\boldsymbol{s})]\frac{\partial \mu_1(\boldsymbol{s})}{\partial \boldsymbol{a}} + \sum_{\boldsymbol{u} \in D_2} [I_2(\boldsymbol{u}) - \mu_2(\boldsymbol{u})]\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}}\right\}. \quad \text{(b4)}$$

As seen from Section II, there is not a simple explicit expression for $\hat{\boldsymbol{a}}$. Rather, $\hat{\boldsymbol{a}}$ must be obtained numerically by minimizing (14), or equivalently, (27). However, it is easy to show that $\hat{\boldsymbol{a}}$ is unbiased for 3-D planar surfaces, and is *asymptotically unbiased* for more general but reasonably smooth 3-D surfaces.

The negative of the expectation of the second derivatives is now computed.

$$-E\left[\frac{\partial^2 \ln p(I_1, I_2 | \alpha)}{\partial \mu(\boldsymbol{t})\,\partial \mu(\boldsymbol{t}')}\right] = \begin{cases} 0 & \text{if } \boldsymbol{t}' \neq \boldsymbol{t}; \\ \dfrac{1}{\sigma^2}[\delta_1(\boldsymbol{t}) + \delta_2(\boldsymbol{t})] & \text{if } \boldsymbol{t}' = \boldsymbol{t}. \end{cases} \quad \text{(b5)}$$

$$-E\left[\frac{\partial^2 \ln p(I_1, I_2 | \alpha)}{(\partial \sigma^2)^2}\right] = \frac{d_1 + d_2}{2\sigma^4}. \quad \text{(b6)}$$

If $d_{12} = d_1 = d_2$, (b6) becomes $d_1\sigma^{-4}$.

$$\begin{aligned} -E\left[\frac{\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{(\partial \boldsymbol{a}_T)^2}\right] &= \frac{1}{\sigma_T^2}\left\{\sum_{\boldsymbol{s} \in D_1}\left[\frac{\partial \mu_1(\boldsymbol{s})}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu_1(\boldsymbol{s})}{\partial \boldsymbol{a}_T}\right] + \sum_{\boldsymbol{u} \in D_2}\left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]\right\} \\ &= \frac{1}{\sigma_T^2}\left\{\sum_{\boldsymbol{s} \in D_1}\left[\frac{\partial \mu(\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu(\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right] + \sum_{\boldsymbol{u} \in D_2}\left[\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]\right\}. \end{aligned} \quad \text{(b7)}$$

There are cross terms. These are seen by direct computation to be

$$-E\left[\frac{\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{\partial \sigma_T^2 \partial \boldsymbol{a}_T}\right] = 0 \quad \text{(b8)}$$

$$-E\left[\frac{\partial^2 \ln p(I_1, I_2 | \alpha_T)}{\partial \mu_T(\boldsymbol{t})\,\partial \boldsymbol{a}_T}\right] = \frac{1}{\sigma_T^2}\left\{\frac{\partial \mu(\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\delta_1(\boldsymbol{t}) + \frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\delta_2(\boldsymbol{t})\right\}. \quad \text{(b9)}$$

Finally, note that $-E[\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)/\partial \mu_T(\boldsymbol{t})\,\partial \sigma_T^2] = 0$.

From the preceding, the Fisher Information Matrix has the form

$$\cdot\left[\begin{array}{ccccc|ccc} \gamma_1 & 0 & \cdot & 0 & 0 & & & \\ 0 & \gamma_2 & \cdot & \cdot & 0 & & H & \\ \cdot & \cdot & \cdot & \cdot & \cdot & & & \\ 0 & 0 & \cdot & \gamma_L & 0 & & & \\ 0 & 0 & \cdot & 0 & \zeta_0 & 0 & \cdot & 0 \\ \hline & & & & 0 & & & \\ & & H^T & & \vdots & & \Theta & \\ & & & & 0 & & & \end{array}\right] \qquad \text{(b10)}$$

where L is the number of those pixels on the object surface that are seen in at least one of the D_1 and D_2,

$$\gamma_i = -E\left[\frac{\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{(\partial \mu_T(t_i))^2}\right], \quad i = 1, 2, \cdots, L,$$

$$\zeta_0 = -E\left[\frac{\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{(\partial \sigma_T^2)^2}\right],$$

and for $\boldsymbol{a}^T$ having K components,

$$\Theta = -E\left[\frac{\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{(\partial \boldsymbol{a}_T)^2}\right],$$

and H has ith row

$$-E\left[\frac{\partial}{\partial \boldsymbol{a}_T}\left[\frac{\partial \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)}{\partial \mu_T(t_i)}\right]\right], \quad i = 1, 2, \cdots, L.$$

Our primary interest is in finding a lower bound for the error covariance matrix for estimating $\boldsymbol{a}_T$. We make use of the following known matrix inversion result

$$F^{-1} = \left[\begin{array}{c|c} S & R \\ \hline G & Q \end{array}\right]^{-1} = \left[\begin{array}{c|c} (S - RQ^{-1}G)^{-1} & -S^{-1}R(Q - GS^{-1}R)^{-1} \\ \hline -Q^{-1}G(S - RQ^{-1}G)^{-1} & (Q - GS^{-1}R)^{-1} \end{array}\right]. \qquad \text{(b11)}$$

Upon making the identification

$$S = \begin{bmatrix} \gamma_1 & 0 & \cdot & 0 & 0 \\ 0 & \gamma_2 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \gamma_L & 0 \\ 0 & 0 & \cdot & 0 & \zeta_0 \end{bmatrix}, \quad R = \left[\begin{array}{c} H \\ \hline 0\ 0\ \cdot\ \cdot\ 0 \end{array}\right],$$

$$G = R^T, \quad Q = \Theta,$$

so that

$$F = \left[\begin{array}{c|c} S & R \\ \hline G & Q \end{array}\right],$$

there results

$$F_a^{-1} = (Q - GS^{-1}R)^{-1}$$

$$= \left\{\frac{1}{\sigma_T^2}\left(\sum_{s \in D_1}\left[\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}))}{\partial \boldsymbol{a}_T}\right]\right.\right.$$

$$\left. + \sum_{u \in D_2}\left[\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T\left[\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]\right)$$

$$- \sum_{s \in D_1 \cup \boldsymbol{h}^{-1}(D_2, \boldsymbol{b}, \boldsymbol{a}_T)}\left(\frac{\sigma_T^2}{(\delta_1(\boldsymbol{t}) + \delta_2(\boldsymbol{t}))}\frac{1}{\sigma_T^4}\right.$$

$$\cdot\left[\delta_1(\boldsymbol{t})\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right.$$

$$\left. + \delta_2(\boldsymbol{t})\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T$$

$$\cdot\left[\delta_1(\boldsymbol{t})\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right.$$

$$\left.\left.\left. + \delta_2(\boldsymbol{t})\frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]\right)\right\}^{-1}. \qquad \text{(b12)}$$

Summing $\boldsymbol{s}$ over $\boldsymbol{h}^{-1}(D_2, \boldsymbol{b}, \boldsymbol{a}_T)$ in the last summation in (b12) is equivalent to summing $\boldsymbol{u}$ over D_2. Equation (b12) simplifies to

$$F_a^{-1} = \frac{\sigma_T^2}{2}\left\{\sum_{s \in \tilde{D}_{12}}\left[\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T} - \frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T\right.$$

$$\left.\cdot\left[\frac{\partial \mu(\boldsymbol{q}_1(s, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T} - \frac{\partial \mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}_2, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]\right\}^{-1}. \qquad \text{(b12a)}$$

Note that due to cancellations, only the subset $\tilde{D}_{12}$ of D_1 contributes to the final expression for the C.R. Bound, (b12a). [$\tilde{D}_{12}$ is defined preceding (35).] This is as expected since data points cannot contribute to the estimation of $\boldsymbol{a}$ unless there is a point in each image that is due to the same point on the object surface. The right side of (b12a), the Cramer-Rao Bound for the estimation error for the parameter vector $\boldsymbol{a}_T$, is interesting and is interpreted in Section III-C.

We emphasize that the surface pattern model used here is that for which the reflectance at each surface point is an *a priori* unknown arbitrary parameter. For this model, the only data that contributes to surface reconstruction are pairs of points, one point in each image where each point in a pair is an image of the same point on the object surface. If other surface pattern models are used, such as contour polynomial models [28], or Markov random field texture models [29], then even if a surface point is seen in only one image, the data point does contribute to the 3-D surface reconstruction.

Appendix B2

Proceeding similarly to the derivation of (b12a), in this appendix the C.R. Bound for $\boldsymbol{a}_T$ is found in terms of $\mu_2(\boldsymbol{u})$ when $\mu_1(\boldsymbol{s})$ is assumed to be the true *a priori* unknown pattern parameters.

We introduce a few identities that are useful for manipulating the expressions that must be dealt with in these multi-image problems. First, how are the spatial gradients of two images related? Since $\mu(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) = \mu_2(\boldsymbol{u})$ where $\boldsymbol{t} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$, it follows that

$$\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} = \frac{\partial \mu(\boldsymbol{t})}{\partial \boldsymbol{t}} \frac{\partial \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})}{\partial \boldsymbol{u}}. \tag{b13}$$

Similarly, since $\mu_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})) = \mu_2(\boldsymbol{u})$ where $\boldsymbol{s} = \boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$, it follows that

$$\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{u}} = \frac{\partial \mu_1(\boldsymbol{s})}{\partial \boldsymbol{s}} \frac{\partial \boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})}{\partial \boldsymbol{u}}. \tag{b14}$$

Similar expressions exist when $\boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a})$ and $\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a})$ are used in place of $\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$ and $\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})$, respectively. Finally, since $\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}) = \boldsymbol{q}_1^{-1}(\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}), \boldsymbol{a})$, and $\boldsymbol{t} = \boldsymbol{q}_1(\boldsymbol{q}_1^{-1}(\boldsymbol{t}, \boldsymbol{a}), \boldsymbol{a})$, it follows that

$$\frac{\partial \boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})}{\partial \boldsymbol{a}} = \frac{\partial \boldsymbol{q}_1^{-1}(\boldsymbol{t})}{\partial \boldsymbol{t}} \frac{\partial \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a})}{\partial \boldsymbol{a}} + \frac{\partial \boldsymbol{q}^{-1}(\boldsymbol{t}, \boldsymbol{a})}{\partial \boldsymbol{a}} \tag{b15}$$

and

$$\frac{\partial \boldsymbol{t}}{\partial \boldsymbol{a}} = 0 = \frac{\partial \boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a})}{\partial \boldsymbol{s}} \frac{\partial \boldsymbol{q}_1^{-1}(\boldsymbol{t}, \boldsymbol{a})}{\partial \boldsymbol{a}} + \frac{\partial \boldsymbol{q}_1(\boldsymbol{s}, \boldsymbol{a})}{\partial \boldsymbol{a}}, \tag{b16}$$

respectively.

We begin with (26). Here, the first summation is not a function of $\boldsymbol{a}$, only the second summation is. Because $\boldsymbol{\mu}_2$ is related to μ_1 through $\mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T)) = \mu_1(\boldsymbol{s})$, and the $\mu_1(\boldsymbol{s})$ are limited to $\boldsymbol{s} \in D_1$, it follows that the only values of $\boldsymbol{s}$ and $\boldsymbol{u}$ that contribute to the C.R. Bound are $\boldsymbol{s} \in \tilde{D}_{12}$ and $\boldsymbol{u} \in \boldsymbol{h}(\tilde{D}_{12}, \boldsymbol{b}, \boldsymbol{a}_T)$ where $\tilde{D}_{12}$ is defined preceding (35). Proceeding as for (b7), we find

$$-E\left[\frac{\partial^2 \ln p(I_1, I_2 | \alpha_T)}{(\partial \boldsymbol{a}_T)^2}\right] = \frac{1}{\sigma_T^2} \sum_{\boldsymbol{u} \in \boldsymbol{h}(\tilde{D}_{12}, \boldsymbol{b}, \boldsymbol{a}_T)} \left\{\left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]^T \left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]\right\}. \tag{b17}$$

Note that when computing $\partial \mu_2(\boldsymbol{u})/\partial \boldsymbol{a}_T$, the dependence of $\mu_2(\boldsymbol{u})$ on $\boldsymbol{a}_T$ is through $\boldsymbol{u} = \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T)$, where $\boldsymbol{s} \in \tilde{D}_{12}$. Also, we have used the fact that $\partial \mu_1(\boldsymbol{s})/\partial \boldsymbol{a}_T = 0$ for all $\boldsymbol{s}$ since the $\mu_1(\boldsymbol{s})$ are assumed to be the true pattern parameters. Furthermore, $-E[\partial^2 \ln p(I_1, I_2 | \boldsymbol{\alpha}_T)/\partial \boldsymbol{a}_T \partial \boldsymbol{\mu}_1]$ is a matrix having row

$$\sigma_T^{-2} \frac{\partial \mu_1(\boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T} \tag{b18}$$

associated with point $\boldsymbol{s} = \boldsymbol{h}^{-1}(\boldsymbol{u}, \boldsymbol{b}, \boldsymbol{a}_T)$. Upon using this and (b17) in equations similar to (b10) and (b11), we find

$$F_a^{-1} = 2\sigma_T^2 \left\{ \sum_{\boldsymbol{u} \in \boldsymbol{h}(\tilde{D}_{12}, \boldsymbol{b}, \boldsymbol{a}_T)} \left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]^T \left[\frac{\partial \mu_2(\boldsymbol{u})}{\partial \boldsymbol{a}_T}\right]\right\}^{-1}. \tag{b19}$$

The explicit dependence on $\boldsymbol{s}$ is exhibited in

$$F_a^{-1} = 2\sigma_T^2 \left\{ \sum_{\boldsymbol{s} \in \tilde{D}_{12}} \left[\frac{\partial \mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]^T \cdot \left[\frac{\partial \mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T))}{\partial \boldsymbol{a}_T}\right]\right\}^{-1}. \tag{b20}$$

If we wish, from (b2), $\partial \mu_2(\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T))/\partial \boldsymbol{a}_T$ can be expressed in terms of $\mu_1(\boldsymbol{s})$ by $(\partial \mu_1(\boldsymbol{s})/\partial \boldsymbol{s})(\partial \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T)/\partial \boldsymbol{s})(\partial \boldsymbol{h}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{a}_T)/\partial \boldsymbol{a}_T)$.

Appendix C
The Plane

We derive the expression for the vector $\partial z/\partial \boldsymbol{a}$ for a plane. Note that there are a number of different sets of parameters that can be used for representing a plane (or a cylinder, or a more general surface). We use the canonical parameterization in this section. We use the equation

$$0 = g(x, y, z) = \beta_1 x + \beta_2 y + \beta_3 z - d \tag{c1}$$

subject to the constraint

$$0 = f(x, y, z) = \beta_1^2 + \beta_2^2 + \beta_3^2 - 1. \tag{c1.1}$$

Note, $|d|$ is the distance from the plane to the origin in this representation. It is assumed that the plane is in general position, because if, e.g., $\beta_3 = 0$, then the plane normal is orthogonal to the first camera's optical axis, and the plane surface is not seen by the first camera since the camera then sees only the plane's edge. Equation (c1.1) can be used to solve for β_3 in terms of β_1 and β_2. Hence, we can take $\boldsymbol{a}$ to be $\boldsymbol{a}^T = (\beta_1, \beta_2, d)$. Now $\partial z/\partial \boldsymbol{a} = -((\partial g/\partial \boldsymbol{a})/(\partial g/\partial z))$. Using (c1.1), we get $\partial \beta_3/\partial \beta_1 = -((\partial f/\partial \beta_1)/(\partial f/\partial \beta_3)) = -2\beta_1/2\beta_3 = -(\beta_1/\beta_3)$. Similarly, $\partial \beta_3/\partial \beta_2 = -((\partial f/\partial \beta_2)/(\partial f/\partial \beta_3)) = -(\beta_2/\beta_3)$. Hence, $\partial g/\partial z = \beta_3$

$$\frac{\partial g}{\partial \boldsymbol{a}} = \begin{bmatrix} \dfrac{\partial g}{\partial \beta_1} = x + z\dfrac{\partial \beta_3}{\partial \beta_1} = x - z\dfrac{\beta_1}{\beta_3} = \dfrac{\beta_3 x - \beta_1 z}{\beta_3} \\ \dfrac{\partial g}{\partial \beta_2} = y + z\dfrac{\partial \beta_3}{\partial \beta_2} = \dfrac{\beta_3 y - \beta_2 z}{\beta_3} \\ \dfrac{\partial g}{\partial d} = -1. \end{bmatrix}$$

Thus,

$$\frac{\partial z}{\partial \boldsymbol{a}} = \left(\frac{\beta_1 z - \beta_3 x}{\beta_3^2}, \frac{\beta_2 z - \beta_3 y}{\beta_3^2}, \frac{1}{\beta_3}\right). \tag{c2}$$

Appendix D
The Cylinder

The canonical parameterization here is

$$\begin{aligned} 0 = g(x, y, z) &= (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \\ &\quad - (\alpha_1 x + \alpha_2 y + \alpha_3 z)^2 - R^2 \\ &\alpha_1 x_0 + \alpha_2 y_0 + \alpha_3 z_0 = 0 \\ &\alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 1. \end{aligned} \tag{d1}$$

The unit vector $(\alpha_1, \alpha_2, \alpha_3)^T$ is in the direction of the cylinder axis, and the second equation in (d1) forces the point (x_0, y_0, z_0) to be the point on the cylinder axis that is closest to the origin. Because of the third equation in (d1), we take two of the three α_i to be independent parameters, and because of the second equation we take two of x_0, y_0, z_0 to be independent parameters. Hence, we choose $\boldsymbol{a} = (x_0, y_0, \alpha_1, \alpha_2, R)^T$. Then

$$\partial\alpha_3/\partial\alpha_1 = -\alpha_1/\alpha_3, \ \partial\alpha_3/\partial\alpha_2 = -\alpha_2/\alpha_3$$

$$\frac{\partial g}{\partial x_0} = -2(x - x_0)$$

$$\frac{\partial g}{\partial y_0} = -2(y - y_0)$$

$$\frac{\partial g}{\partial \alpha_1} = -2(\alpha_1 x + \alpha_2 y + \alpha_3 z)\left(x + z\frac{\partial \alpha_3}{\partial \alpha_1}\right)$$

$$\frac{\partial g}{\partial \alpha_2} = -2(\alpha_1 x + \alpha_2 y + \alpha_3 z)\left(y + z\frac{\partial \alpha_3}{\partial \alpha_2}\right)$$

$$\frac{\partial g}{\partial R} = -2R$$

$$\frac{\partial g}{\partial z} = 2(z - z_0) - 2\alpha_3(\alpha_1 x + \alpha_2 y + \alpha_3 z).$$

Denote $c = [(z - z_0) - \alpha_3(\alpha_1 x + \alpha_2 y + \alpha_3 z)]$, and $d = \alpha_1 x + \alpha_2 y + \alpha_3 z$. Then

$$\begin{aligned} \frac{\partial z}{\partial \boldsymbol{a}} = \big(&(x - x_0)/c, (y - y_0)/c, d(\alpha_3 x - \alpha_1 z)/c, \\ &d(\alpha_3 y - \alpha_2 z)/c, R/c\big). \end{aligned} \tag{d2}$$

Appendix E
The General Quadric Surface

The general quadric is given by

$$\begin{aligned} 0 = g(x, y, z) &= a_{11}x^2 + 2a_{12}xy + 2a_{13}xz \\ &\quad + a_{22}y^2 + 2a_{23}yz + a_{33}z^2 + 2a_{41}x \\ &\quad + 2a_{42}y + 2a_{43}z + a_{44}. \end{aligned} \tag{e1}$$

As can be seen in (e1), the parameter values for a surface are not unique, as multiplication of all coefficients by the same arbitrary constant leaves the equation unchanged. Hence, a constraint must be imposed. One that results in quadric surface estimation that is invariant to the choice of origin location and axis orientation for the coordinate system used for describing the object is [1]

$$0 = a_{11}^2 + 2a_{12}^2 + 2a_{13}^2 + a_{22}^2 + 2a_{23}^2 + a_{33}^2 - 2. \tag{e1.1}$$

Using (e1.1), we arbitrarily choose the dependent parameter to be a_{33}. We assume that at least one of a_{13}, a_{23}, a_{33}, and a_{43} is nonzero. Otherwise, the surface would not be seen as we would have a ruled surface parallel to the z-axis. Let $\boldsymbol{a}^T = (a_{11}, a_{12}, a_{13}, a_{22}, a_{23}, a_{41}, a_{42}, a_{43}, a_{44})$. Upon using (e1) and (e1.1), we obtain

$$\frac{\partial g}{\partial z} = 2a_{33}z + 2a_{43}$$

$$\frac{\partial g}{\partial a_{11}} = x^2, \ \frac{\partial g}{\partial a_{12}} = 2xy, \ \frac{\partial g}{\partial a_{13}} = 2xz$$

$$\frac{\partial g}{\partial a_{22}} = y^2, \ \frac{\partial g}{\partial a_{23}} = 2yz, \ \frac{\partial g}{\partial a_{33}} = z^2$$

$$\frac{\partial g}{\partial a_{41}} = 2x, \ \frac{\partial g}{\partial a_{42}} = 2y, \ \frac{\partial g}{\partial a_{43}} = 2z, \ \frac{\partial g}{\partial \mathrm{a}_{44}} = 1.$$

Use of the preceding and $\partial z/\partial \boldsymbol{a} = -((\partial g/\boldsymbol{a})/(\partial g/\partial z))$ leads to

$$\begin{aligned} \frac{\partial z}{\partial \boldsymbol{a}} &= -(2a_{33}z + 2a_{43})^{-1} \\ &\quad \cdot (x^2, 2xy, 2xz, y^2, 2yz, 2x, 2y, 2z, 1). \end{aligned} \tag{e2}$$

Note that a_{33} can be obtained by solving (e1.1), and z can be obtained by solving (e1).

References

[1] B. Cernuschi-Frias, "Orientation and location parameter estimation of quadric surfaces in 3-D space from a sequence of images," Ph.D. Thesis, Brown Univ., Division of Eng., May 1984. Also Tech. Rep. #LEMS-5, Feb. 1984.

[2] B. Cernuschi-Frias, P. N. Belhumeur, and D. B. Cooper, "Estimating and recognizing parameterized 3-D objects using a moving camera," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, San Francisco, CA, June 1985, pp. 167–171.

[3] F. G. Amblard, D. B. Cooper, and B. Cernuschi-Frias, "Estimation by multiple views of outdoor terrain modeled by stochastic processes," in *Proc. 5th SPIE Cambridge Conf. Intell. Robots Comput. Vision*, vol. 726, Cambridge, MA, Oct. 27–31, 1986, pp. 36–45.

[4] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 139–154, Mar. 1985.

[5] R. C. Bolles and H. H. Baker, "Epipolar-plane image analysis: A technique for analyzing motion sequence," *Proc. Image Understanding*, pp. 137–148, Dec. 1985.

[6] W. T. Miller III, "Video image matching using phase-locked loop techniques," in *Proc. IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, Apr. 7-10, 1986 , pp. 112-117.

[7] S. Castan and J. Shen, "A stereo vision algorithm taking into account the perspective distortions," in *Proc. 7th Int. Conf. Pattern Recogn.*, Montreal, Canada, July 30-Aug. 3, 1984, pp. 444-446.

[8] O. D. Faugeras, N. Ayache, and B. Faverjon, "Building visual maps by combining noisy stereo measurements," in *Proc. 1986 IEEE Conf. Robot. Automat.*, San Francisco, CA, Apr. 7-10, 1986, pp. 1433-1438.

[9] M. Drumheller and T. Poggio, "On parallel stereo," in *Proc. 1986 IEEE Conf. Robot. Automat.*, San Francisco, CA, Apr. 7-10, 1986, pp. 1439-1448.

[10] A. M. Waxman and K. Wohn, "Image flow theory: A framework for 3-D inference from time-varying imagery," in *Advances In Computer Vision*, vol. I, C. Brown, Ed. Hillsdale, NJ: Erlbaum, 1988, ch. 3, pp. 165-224.

[11] Y. P. Hung, B. Cernuschi-Frias, and D. B. Cooper, "Maximum likelihood estimation of parameterized surfaces in three dimensional space using a moving camera," in *Proc. JPL Cal. Tech. Workshop Space Telerobotics*, Jan. 20-22, 1987, pp. 71-83; also Tech. Rep. LEMS-33, Brown Univ., Dec. 1986.

[12] R. M. Bolle and D. B. Cooper, "On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 619-638, Sept. 1986.

[13] H. L. van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: Wiley, 1968, pp. 74-85.

[14] S. S. Wilks, *Mathematical Statistics*. New York: Wiley, 1963, p. 419.

[15] D. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973, p. 381.

[16] A. E. Albert and L. A. Gardner, *Stochastic Approximation and Nonlinear Regression*. Cambridge, MA: The M.I.T. Press, 1967.

[17] B. Cernuschi-Frias, D. B. Cooper, P. Belhumeur, and Y. P. Hung, "Toward a Bayesian theory for estimating and recognizing parameterized 3-D objects using two or more images taken from different positions," Division of Eng., Brown Univ., Tech. Rep. LEMS-32, Dec. 1986.

[18] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice Hall, 1982.

[19] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *Comput. Surveys*, vol. 17, no. 1, pp. 75-145, Mar. 1985.

[20] R. M. Bolle and D. Sabbah, "Depth map processing for recognizing objects modelled by planes and quadrics of revolution," in *Proc. Intell. Autonomous Syst.*, Elsevier Science Publishers, B. V., Amsterdam, Dec. 1986, pp. 142-150.

[21] D. R. Cunningham, R. D. Laramore, and E. Barrett, "Detection in image dependent noise," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 603-610, Sept. 1976.

[22] D. B. Cooper, "Maximum likelihood estimation of Markov-process blob boundaries in noisy images," *IEEE Trans. Pattern Anal. Machine Intell.*, Oct. 1979, pp. 68-79. Reprinted with corrections in *Digital Image Processing and Analysis: Vol. 2: Digital Image Analysis*, R. Chellappa and A. A. Sawchuk editors, IEEE Computer Soc. Press, 1985.

[23] D. B. Cooper, H. Elliott, F. Cohen, L. Reiss, and P. Symosek, "Stochastic boundary estimation and object recognition," in *Image Modeling*, A. Rosenfeld, Ed. New York: Academic, 1981, pp. 63-94.

[24] B. K. P. Horn, *Robot Vision*. New York: McGraw-Hill, 1986.

[25] R. D. Eastman and A. M. Waxman, "Using disparity functionals for stereo correspondence and surface reconstruction," Cent. Automat. Res., Univ. Maryland, Tech. Rep. CS-TR-1547, Oct. 1985.

[26] J. F. Silverman and D. B. Cooper, "Bayesian clustering for unsupervised estimation of surface and texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 482-495, July 1988.

[27] Y. P. Hung, D. B. Cooper, and B. Cernuschi-Frias, "Bayesian estimation of 3-D surfaces from a sequence of images," in *1988 IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, Apr. 24-29, 1988, pp. 906-911.

[28] D. B. Cooper, Y. P. Hung, and G. Taubin, "A new model-based stereo approach for 3D surface reconstruction using contours on the surface pattern," in *Proc. 1988 Int. Conf. Comp. Vision*, Tarpon Springs, FL, Dec. 1988, pp.74-83.

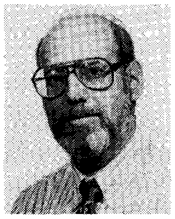
[29] F. S. Cohen and D. B. Cooper, "A decision theoretic approach for 3-D vision," in *Proc. IEEE Comp. Soc. Conf. Comput. Vision Pattern Recogn.*, June 5-9, 1988, Ann Arbor, MI, pp. 964-972.

[30] D. G. Hakala, R. C. Hillyard, P. F. Malraison, and B. F. Nource, "Natural quadrics in mechanical design," in *SIGGRAPH '81, Seminal: Solid Modelling*, Dallas, TX, Aug. 1981.

Bruno Cernuschi-Frias (S'77-M'78) was born in Montevideo, Uruguay, on April 7, 1952. He received the Ingeniero Electromecánico Orientación Electrónica degree from the Facultad de Ingeniería de la Universidad de Buenos Aires (FIUBA), Buenos Aires, Argentina, in 1976, and the M.Sc. and Ph.D degrees in electrical engineering from Brown University, Providence, RI, in 1983 and 1984, respectively.

During 1977-1980 he was engaged in research and teaching at FIUBA. A leave of absence during January 1981-December 1983 was spent on graduate study and research at Brown University in the field of computer vision. He was a Visiting Professor at Brown University from March to June 1986. Since 1984 he has been at FIUBA as a member of the Scientific Research Career of the CONICET where he conducts research in computer vision.

Dr. Cernuschi-Frías is a member of the Association for Computing Machinery, AAAS, Sigma Xi, and the New York Academy of Sciences. He has received fellowships from the Organization of American States, Faculatad de Ingeniería, Universidad de Buenos Aires, and from IBM (Thomas J. Watson Research Center).



David B. Cooper (S'53-M'64) received the B.Sc. and Sc.M. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, under the industrial co-op program in January 1957, and the Ph.D. degree in applied mathematics from Columbia University, New York, NY, in June 1966.

From 1957 to 1966, he worked first for Sylvania Electric Products, Inc., Mountain View, CA, and then for the Raytheon Company, Waltham, MA, on communication and radar system analysis. Since 1966 he has been a Professor of Engineering at Brown University, Providence, RI. He held an M.I.T. Summer Overseas Fellowship at Marconi Wireless, England, in 1955, was a research staff member of the Technische Hogeschool, Delft, The Netherlands, for the 1972-1973 academic year and was a visiting researcher at the Institute National de Recherche en Informatique et en Automatique, near Paris, for the Fall 1988. His research interests are in computer vision, pattern recognition and applied stochastic processes.

Yi-Ping Hung (S'89) received the B.S.E.E. degree from National Taiwan University in 1982, the Sc.M. degree in electrical engineering and the Sc.M. degree in applied mathematics, both from Brown University, in 1987 and 1988, respectively. He is currently working toward the Ph.D. degree in the Division of Engineering at Brown University.

He has conducted research on computer vision at the Laboratory for Engineering Man/Machine Systems (LEMS), Brown University, for almost five years. He was employed by Philips Laboratories, Briarcliff Manor, NY, during the Summers of 1987 and 1988, working on robot vision. His current research interests include computer vision, robotics, machine learning, parallel computation, and neural networks.

Peter N. Belhumeur (S'85) was born in Providence, RI, on August 24, 1963. As a student at Brown University, he developed computer vision algorithms in the LEMS Laboratory. He received an Sc.B. from Brown University, Providence in 1985.

After graduation he worked for Analog Devices creating an optical character recognition system. In January 1987, he was contracted by AT&T to write a large-scale database network. At the completion of this project, he spent several months working for the Coalition for the Homeless in New York City, before moving to Cambridge, Massachusetts. He currently works for Harvard Medical School's Program for the Analysis of Clinical Strategies.